



TUGAS AKHIR - TE 141599

**ANALISIS UNJUK KERJA INTERKONEKSI IPV6 DAN
IPV4 DENGAN METODE IPV6 TUNNEL BROKER**

M. Yusro Muhtadi
NRP 2210 100 155

Dosen Pembimbing
Dr. Ir. Achmad Affandi, DEA
Ir. Djoko Suprajitno Rahardjo, MT

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015

Halaman ini sengaja dikosongkan



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 141599

**PERFORMANCE ANALYSIS IPV6-IPV4
INTERCONNECTION USING IPV6 TUNNEL BROKER
METHOD**

M. Yusro Muhtadi
NRP 2210 100 155

Advisor
Dr. Ir. Achmad Affandi, DEA
Ir. Djoko Suprajitno Rahardjo, MT

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2015

Halaman ini sengaja dikosongkan

ANALISIS UNJUK KERJA INTERKONEKSI IPV6 DAN IPV4 DENGAN METODE IPV6 TUNNEL BROKER

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada**

**Bidang Studi Telekomunikasi Multimedia
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Mengetahui/Menyetujui:

Dosen Pembimbing I,

Dosen Pembimbing II,

Dr. Ir. Achmad Affandi, DEA
NIP.1965 10/14 1990 02 1001

Ir. Djoko Suprajitno Rahardjo, MT.
NIP.1955 06 22 1987 01 1001



**SURABAYA
JANUARI, 2015**

ANALISIS UNJUK KERJA INTERKONEKSI IPV6 DAN IPV4 DENGAN METODE IPV6 TUNNEL BROKER

M. Yusro Muhtadi
2210 100 155

Dosen Pembimbing I : Dr. Ir. Achmad Affandi, DEA
Dosen Pembimbing II : Ir. Djoko Suprajitno Rahardjo, MT.

Abstrak:

Dalam mekanisme transisi IPv6 dan IPv4 penggunaan metode *tunneling* cukup populer. Alternatif yang dapat digunakan adalah menggunakan *IPv6 tunnelbroker*. *Tunnelbroker* akan menyediakan konfigurasi untuk melakukan *tunneling* IPv6 melalui jaringan IPv4 kepada *user* yang terhubung ke jaringan internet. *Tunnelbroker* akan membentuk, memodifikasi dan menghapus *tunnel* sesuai dengan permintaan *user*. Tujuan dari tugas akhir ini untuk mengetahui kinerja jaringan pada IPv6 *tunnelbroker* yang akan dilalui paket menggunakan aplikasi *file transfer protocol*. File yang terdiri dari berbagai macam jenis ukuran dan tipe akan dilewatkan melalui *tunnel*, yang ada. Hasil yang didapatkan, untuk file yang berukuran kecil yaitu file(a).txt berukuran 166,790 KB *latency*nya pada jaringan IPv6 *tunnel broker* akan lebih besar 9,77% dibanding jaringan IPv4 dan 13,93% dibandingkan dengan IPv6. Sedangkan *throughput* dari IPv6 *tunnel broker* lebih besar 13,4% terhadap IPv4 dan lebih kecil 13,47% dari IPv6. Untuk file yang berukuran besar yaitu file(e).iso sebesar 51.013,632 KB *latency* pada jaringan IPv6 *tunnel broker* akan lebih besar 210,26% dibandingkan jaringan IPv4 dan 291,6% dibandingkan dengan IPv6. Sedangkan *throughput* dari IPv6 *tunnel broker* lebih kecil 63,75% terhadap IPv4 dan 71,28% dari IPv6. Selain itu dapat ditarik kesimpulan bahwa jenis tipe file tidak berpengaruh terlalu besar terhadap performa yang ada, namun ukuran file yang menjadi faktor terbesar dalam performa yang ada. Selain itu nilai *latency* akan berbanding terbalik dengan nilai *throughput* dikarenakan *throughput* adalah hasil dari pembagian ukuran file dengan *latency*.

Kata Kunci : IPv6, *tunnelbroker*, *file transfer protocol*

Halaman ini sengaja dikosongkan

PERFORMANCE ANALYSIS IPV6-IPV4 INTERCONNECTION USING IPV6 TUNNEL BROKER METHOD

M. Yusro Muhtadi
2210 100 155

1st Lecturer : Dr. Ir. Achmad Affandi, DEA
2nd Lecturer : Ir. Djoko Suprajitno Rahardjo, MT.

Abstract :

The tunneling method is quite popular among IPv4 to IPv6 transition mechanisms. An alternative that could be used is the IPv6 tunnel broker. *Tunnel broker* will provide a configuration to tunnel IPv6 over IPv4 networks to the user connected to the internet. Tunnel broker will form, modify, and erase tunnels according to the users demand. The goal of this final project is to find out the performance of the IPv6 tunnel broker which will be burdened with a File Transfer Protocol application. Multiple files which consists of varying sizes and formats will be sent through the tunnel. results shows that for the smallest file which is file(a).txt with the size of 166,790 KB, the latency on the IPv6 tunnel broker network is 9,77% larger than on the IPv4 network and 13,93% larger than the IPv6 network. Where as the throughput on the IPv6 tunnel broker network is larger 13,4% then the IPv4 network and 13,47% smaller than the IPv6 network. For the biggest file which is file(e).iso with the size of 51.013,632 KB, the latency in the IPv6 tunnel broker network is 210,26% larger than the IPv4 network and 291,62% larger then the IPv6 network. Where as the troughput on IPv6 tunnel broker is 63,75% smaller then the IPv4 network and 71,28% smaller than the IPv6 network. It is concluded that the format of the file does not significantly effect the network performance, however the size of the file becomes a major factor to it and the value of it's latency is inversly proportional to it's throughput because the throughput is file's size devided by the latency.

Key Word:IPv6, tunnel broker, file transfer protocol

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan Tugas Akhir ini dengan baik karena rahmat Allah SWT. Puji dan syukur senantiasa dipanjatkan kepada Allah SWT. Serta tak lupa pula shalawat dan salam dihaturkan kepada Nabi Muhammad SAW, keluarga, sahabat, dan pengikut beliau hingga akhir zaman.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan dalam menyelesaikan pendidikan Strata-1 pada Bidang Studi Telekomunikasi Multimedia, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

“Analisis Unjuk Kerja Interkoneksi IPv6 dan IPv4 dengan Metode IPv6 Tunnel Broker”

Dengan selesainya Tugas Akhir ini, saya mengucapkan terimakasih kepada:

1. Bapak Tri Arief Sardjono, sebagai Ketua Jurusan Teknik Elektro ITS.
2. Bapak A. Affandi dan Bapak Djoko Suprajitno R. sebagai dosen pembimbing serta Bapak Istas Pratomo yang telah banyak memberikan motivasi, meluangkan waktu, dan memberikan saran serta perbaikan kepada penulis.
3. Keluarga penulis, Mama Rochmiyati dan Abah M. Yusni yang selalu mendoakan dan memberikan dukungan hingga penulis dapat menyelesaikan tugas akhir ini.
4. Bapak dan Ibu dosen JTE yang telah memberikan banyak ilmu kepada penulis selama berkuliah.
5. Seluruh teman-teman angkatan E50 yang telah banyak memberikan dukungan serta doa.

Harapan penulis dengan selesainya penyusunan buku tugas akhir ini dapat memberikan informasi yang bermanfaat bagi pembacanya. Terutama bagi mahasiswa Teknik Elektro bidang studi Telekomunikasi Multimedia.

Surabaya, Januari 2015

M. Yusro Muhtadi

DAFTAR ISI

HALAMAN JUDUL INDONESIA	i
HALAMAN JUDUL INGGRIS	iii
PERNYATAAN KEASLIAN	v
LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
BAB 1	1
1.1 Latar Belakang.....	1
1.2 Rumusan Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi.....	2
1.6 Sistematika Penulisan	3
1.7 Relevansi.....	4
BAB 2	5
2.1 Pengenalan Jaringan.....	5
2.1.1 Jaringan <i>Local Area Network</i> (LAN).....	5
2.1.2 Jaringan <i>Metropolitan Area Network</i> (MAN)	6
2.1.3 Jaringan <i>Wide Area Network</i> (WAN)	7
2.2 Referensi Model Open System <i>Interconnection</i> (ISO)	7
2.2.1 <i>Physical Layer</i>	8
2.2.2 <i>Data Link Layer</i>	8
2.2.3 <i>Network Layer</i>	9
2.2.4 <i>Transport Layer</i>	9
2.2.5 <i>Session Layer</i>	10
2.2.6 <i>Presentation Layer</i>	10
2.2.7 <i>Application Layer</i>	10
2.3 Enkapsulasi	11
2.4 File Transfer Protocol (FTP).....	12
2.5 IPv6.....	13
2.4.1 Perubahan dari IPv4 ke IPv6	13
2.4.1.1 Kapasitas Perluasan Alamat.....	13

2.4.1.2	Penyederhanaan Format <i>Header</i>	14
2.4.1.3	<i>Option</i> dan <i>Extension Header</i>	15
2.4.1.4	Kemampuan Pelabelan Aliran Paket.....	15
2.4.1.5	Kemampuan Autentifikasi dan Privasi.....	15
2.4.2	Pengalamatan IPv6.....	16
2.4.2.1	<i>Unicast Address (one-to-one)</i>	16
2.4.2.2	<i>Multicat Address (one-to-many)</i>	16
2.4.2.3	<i>Anycast Address</i>	17
2.4.3	Representasi Alamat IPv6	17
2.6	Mekanisme Transisi IPv6	18
2.5.1	<i>Dual Stack</i>	19
2.5.2	<i>Tunneling</i>	19
2.5.3	Translasi	22
2.7	IPv6 Tunnel Broker	23
2.6.1	<i>Tunnel Broker</i>	24
2.6.2	<i>Tunnel Server</i>	25
2.6.3	Domain Name Service (DNS) Server	25
2.6.4	Mekanisme <i>Tunnel Broker</i>	25
BAB 3		27
3.1	Parameter Simulasi.....	28
3.2	Perangkat Pendukung	29
3.2.1	<i>Hardware</i> (Perangkat Keras).....	29
3.2.1.1	<i>Personal Computer Server</i>	29
3.2.1.2	<i>Personal Computer Client</i>	29
3.2.1.3	<i>Router</i>	30
3.2.2	<i>Software</i> (Perangkat Lunak)	30
3.2.2.1	Windows Operating System (OS)	31
3.2.2.2	Ubuntu Linux	31
3.2.2.3	Cisco IOS Image	31
3.2.2.4	Script PHP.....	32
3.2.2.5	Apache Web Server	32
3.2.2.6	MySQL	32
3.2.2.7	Putty.....	32
3.2.2.8	TFTP Server.....	33
3.2.2.9	FTP Server	33
3.2.2.10	FTP Client	33
3.2.2.11	Wireshark.....	34
3.3	Instalasi IOS Router	34
3.4	Perancangan <i>Tunnel Broker</i>	38

3.4.1 Pembuatan <i>Script</i> PHP	38
3.4.2 Pembuatan <i>Database</i>	39
3.5 Jaringan IPv6 Tunnel Broker	40
3.5.1 Topologi Jaringan	40
3.5.2 Konfigurasi	42
3.6 Jaringan IPv4 Murni	42
3.6.1 Topologi Jaringan	42
3.6.2 Konfigurasi	43
3.7 Jaringan IPv6 Murni	44
3.7.1 Topologi Jaringan	44
3.7.2 Konfigurasi	45
3.8 Pembentukan Server dan Client FTP	45
3.9 Metode Pengambilan Data.....	47
BAB 4.....	49
4.1 Analisis Jaringan.....	49
4.1.1 Analisis Jaringan IPv6 Tunnel Broker	49
4.1.2 Analisis Jaringan IPv4.....	52
4.1.3 Analisis Jaringan IPv6.....	53
4.2 Tampilan Web Tunnel Broker.....	55
4.3 Analisis Jaringan IPv6 Tunnel Broker	57
4.3.1 Analisis Latency Jaringan IPv6 Tunnel Broker	57
4.3.2 Analisis Troughput Jaringan IPv6 Tunnel Broker	61
4.4 Analisis Jaringan IPv4	64
4.4.1 Analisis Latency Jaringan IPv4	64
4.4.2 Analisis Troughput Jaringan IPv4	67
4.5 Analisis Jaringan IPv6	70
4.5.1 Analisis Latency Jaringan IPv6	70
4.5.2 Analisis Troughput Jaringan IPv6	73
4.6 Analisis Perbandingan Parameter.....	76
4.6.1 Analisis Perbandingan Latency Ukuran Berbeda	76
4.6.2 Analisis Perbandingan Troughput Ukuran Berbeda	78
4.7 Analisis Keseluruhan	80
4.8 Strategi Implementasi Tunnel Broker	81
BAB 5.....	85
5.1 Kesimpulan.....	85
5.2 Saran.....	86
DAFTAR PUSTAKA	87
Lampiran A	89
Lampiran B.....	101

RIWAYAT HIDUP 113

DAFTAR GAMBAR

Gambar 2.1 Jaringan LAN	6
Gambar 2.2 Jaringan MAN	6
Gambar 2.3 Model ISO 7 layer	7
Gambar 2.4 Proses enkapsulasi	11
Gambar 2.5 Perbandingan format <i>header</i> IPv4 dan IPv6	14
Gambar 2.6 Metode <i>dual stack</i>	19
Gambar 2.7 Metode <i>tunneling</i>	20
Gambar 2.8 Proses enkapsulasi paket mekanisme <i>tunneling</i>	22
Gambar 2.9 Metode NAT-PT.....	23
Gambar 2.10 Arsitektur IPv6 <i>tunnel broker</i>	24
Gambar 2.11 Mekanisme <i>tunnel broker</i>	25
Gambar 3.1 Diagram alir perancangan dan implementasi sistem	27
Gambar 3.2 Tampilan show version	35
Gambar 3.3 Keterangan kapasitas <i>memory router</i>	35
Gambar 3.4 Konfigurasi <i>upgrade IOS</i>	36
Gambar 3.5 Diagram alir <i>database tunnel broker</i>	38
Gambar 3.6 <i>Database user</i>	39
Gambar 3.7 Basis Data IPv6	40
Gambar 3.8 Topologi IPv6 <i>tunnel broker</i>	41
Gambar 3.9 Topologi IPv4 murni.....	43
Gambar 3.10 Topologi IPv6 murni	44
Gambar 3.11 Tampilan awal Filezilla Server.....	46
Gambar 3.12 Filezilla Server <i>ready</i>	46
Gambar 3.13 Filezilla Client	47
Gambar 4.1 Ping host1 ke <i>tunnel broker</i>	49
Gambar 4.2 Traceroute host1 ke <i>tunnel broker</i>	50
Gambar 4.3 Ping jaringan IPv6 <i>tunnel broker</i>	51
Gambar 4.4 Traceroute jaringan IPv6 <i>tunnel broker</i>	51
Gambar 4.5 Ping jaringan IPv4	52
Gambar 4.6 Traceroute jaringan IPv4.....	53
Gambar 4.7 Ping jaringan IPv6	54
Gambar 4.8 Traceroute jaringan IPv6.....	54
Gambar 4.9 Tampilan <i>home tunnel broker</i>	55
Gambar 4.10 Tampilan daftar baru	56
Gambar 4.11 Tampilan status pengguna	56
Gambar 4.12 Tampilan konfigurasi	57
Gambar 4.13 <i>Latency IPv6 tunnel broker</i>	59

Gambar 4.14	<i>Latency IPv6 tunnel broker berdasarkan tipe file.....</i>	60
Gambar 4.15	<i>Troughput IPv6 tunnel broker.....</i>	62
Gambar 4.16	<i>Troughput Ipv6 tunnel broker berdasarkan tipe file.....</i>	63
Gambar 4.17	<i>Latency jaringan IPv4</i>	65
Gambar 4.18	<i>Latency IPv4 berdasarkan tipe file</i>	66
Gambar 4.19	<i>Troughput jaringan IPv4</i>	68
Gambar 4.20	<i>Troughput IPv4 berdasarkan tipe file</i>	69
Gambar 4.21	<i>Latency jaringan IPv6</i>	71
Gambar 4.22	<i>Latency IPv6 berdasarkan tipe file</i>	72
Gambar 4.23	<i>Troughput jaringan IPv6</i>	74
Gambar 4.24	<i>Troughput IPv6 berdasarkan tipe file</i>	75
Gambar 4.25	<i>Rata-rata latency.....</i>	77
Gambar 4.26	<i>Rata-rata troughput.....</i>	79

DAFTAR TABEL

Tabel 2.1 Keterangan <i>header</i> IPv6	14
Tabel 2.2 Perbandingan IPv4 dan IPv6.....	15
Tabel 3.1 Alamat Jaringan IPv6 <i>tunnel broker</i>	41
Tabel 4.1 <i>Latency</i> IPv6 <i>tunnel broker</i> (detik)	58
Tabel 4.2 <i>Latency</i> IPv6 <i>tunnel broker</i> berdasarkan tipe file (detik)	60
Tabel 4.3 <i>Troughput</i> IPv6 <i>tunnel broker</i>	61
Tabel 4.4 <i>Troughput</i> IPv6 <i>tunnel broker</i> berdasarkan tipe file (Kbytes/s)	63
Tabel 4.5 <i>Latency</i> jaringan IPv4 (detik).....	64
Tabel 4.6 <i>Latency</i> IPv4 murni berdasarkan tipe file (detik).....	66
Tabel 4.7 <i>Troughput</i> jaringan IPv4.....	67
Tabel 4.8 <i>Troughput</i> IPv4 murni berdasarkan tipe file (Kbytes/s).....	69
Tabel 4.9 <i>Latency</i> jaringan IPv6 (detik).....	70
Tabel 4.10 <i>Latency</i> IPv6 murni berdasarkan tipe file (detik).....	72
Tabel 4.11 <i>Troughput</i> jaringan IPv6.....	73
Tabel 4.12 <i>Troughput</i> IPv6 murni berdasarkan tipe file (Kbytes/s)	75
Tabel 4.13 Rata-rata <i>latency</i> (detik)	76
Tabel 4.14 Persentase <i>latency</i> antar jaringan	77
Tabel 4.15 Rata-rata <i>troughput</i>	78
Tabel 4.16 Persentase <i>troughput</i> antar jaringan	79

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pemanfaatan teknologi komputer yang semakin berkembang sangat pesat turut mempengaruhi perkembangan teknologi yang ada didalamnya. Hal ini selaras dengan semakin berkurangnya kebutuhan sistem pengalamatan jaringan komputer didalam protokol jaringan TCP/IP yang dinamakan *Internet Protocol Address (IP Address)*. Untuk saat ini, pemanfaatan *IP Address* masih menggunakan generasi *Internet Protocol version 4* atau biasa disebut IPv4. IPv4 sendiri memiliki panjang total 32 bit dan secara teori dapat melakukan pengalamatan 4 miliar perangkat komputer atau lebih tepatnya sebesar 4.294.967.296 yang didapatkan dari 256 (8 bit) dipangkatkan 4 (4 oktet) sehingga nilai maksimum kapasitas IPv4 adalah 255.255.255.255. Karena nilai dihitung dari nol maka nilai host yang dapat difasilitasi adalah $256 \times 256 \times 256 \times 256 = 4.294.967.296$ host. Ipv4 sendiri sudah ditemukan sejak tahun 1970-an dan perangkat yang ada sampai sekarang sudah mencapai miliaran, sehingga hal tersebut mengakibatkan kapasitas host menipis. Untuk mengatasi hal tersebut dikembangkanlah IP generasi terbaru yaitu IPv6.

IPv6 memiliki kapasitas *address* 340 *undencillion* alamat publik, penyusunan alamat lebih terstruktur yang memungkinkan internet terus berkembang dan menyediakan *routing* baru yang tidak terdapat pada IPv4[1]. IPv6 dilengkapi sebuah mekanisme penggunaan *address* secara lokal sehingga dapat dilakukan instalasi secara *plug & play*, aliran data yang *realtime*, mobilitas dari host yang ada, *end-to-end security*, dan juga konfigurasi secara otomatis. Dibalik keunggulan tersebut, metode transisi tidak bisa dilakukan dengan mudah. Namun harus melalui berbagai metode yang ada seperti Dual Stack, Tunnelling dan Translasi.

Dalam tugas akhir ini akan dilakukan metode interkoneksi IPv4 dan IPv6 dengan cara *tunnel broker* yang pada penerapannya *tunnel* akan diaktifkan dan hasil dari koneksi antar jaringan akan diukur berdasarkan ukuran paket, tipe file dan jumlah file yang dikirim.

1.2 Rumusan Permasalahan

Permasalahan yang harus dihadapi adalah mekanisme transisi IPv4 dan IPv6 yang dalam penerapannya harus secara bertahap dan mekanisme *tunnel broker* dirasa metode yang tepat dalam mengisi masa transisi ini. Maka *tunnel broker* perlu diamati kualitasnya, khususnya pada aplikasi yang cukup populer yaitu FTP.

1.3 Batasan Masalah

Batasan masalah yang akan dibahas dalam tugas akhir ini antara lain sebagai berikut :

1. Pengujian dan perbandingan kinerja pada jaringan IPv6 *tunnel broker* berdasarkan ukuran file yang ada dengan jaringan IPv4 murni dan IPv6 murni.
2. Parameter pengukuran yang dianalisis adalah *latency* dan *throughput*
3. Pengujian berjalan dalam *file transfer protocol (FTP)*.

1.4 Tujuan

Adapun tujuan dari pembuatan tugas akhir ini adalah sebagai berikut :

1. Membuat jaringan interkoneksi IPv6 *tunnel broker*.
2. Menganalisis kinerja dari jaringan *tunnel broker* yang sudah terbentuk.

1.5 Metodologi

Tugas akhir ini akan diselesaikan lewat beberapa tahap, yaitu studi literatur, analisis kebutuhan penelitian, perancangan sistem, implementasi sistem, pengambilan dan analisis data, penarikan kesimpulan dan penulisan buku tugas akhir

1. Studi Literatur

Tahap pertama yang dilakukan adalah pengumpulan literatur yang berhubungan dengan topik tugas akhir berupa paper maupun jurnal tentang interkoneksi IPv6 khususnya *tunnel broker*.

2. Analisis Kebutuhan Penelitian

Setelah memahami konsep dasar tugas akhir melalui studi literatur, tahapan selanjutnya adalah menganalisis kebutuhan dari pembangunan jaringan *tunnel broker* baik *software* maupun *hardware*.

3. Perancangan Sistem

Sebelum mengimplementasikan secara langsung, sistem harus dirancang terlebih dahulu agar saat sistem berlangsung bisa berjalan sesuai skenario yang ada.

4. Implementasi Sistem

Pada tahap ini setelah dilakukan rancangan sistem, maka akan langsung dilakukan implementasi dan penyesuaian dengan kondisi rancangan sistem.

5. Pengambilan dan Analisis Data

Pada tahap ini dilakukan pengambilan data dari hasil simulasi tersebut. Data yang diperoleh akan dilakukan analisis terhadap skenario yang ada.

6. Penarikan Kesimpulan

Kesimpulan diperoleh dari hasil analisis yang telah dilakukan berdasarkan data dari sistem.

7. Penulisan Buku Tugas Akhir

Ini adalah tahap akhir dari proses pengerjaan tugas akhir. Didalam penulisan ini akan mencakup semua proses pengerjaan tugas akhir, mulai dari dasar teori yang digunakan hingga penarikan kesimpulan serta saran maupun rekomendasi yang dihasilkan dari penelitian. Selain itu, dibuat pula proseding sebagai ringkasan dan materi tugas akhir berupa slide presentasi. Selanjutnya dilakukan mekanisme pengesahan yang meliputi pengajuan tanda tangan, *draft* buku, buku, dan proseding tugas akhir.

1.6 Sistematika Penulisan

Laporan penelitian tugas akhir ini disusun secara sistematis dibagi dalam beberapa bab, dengan rincian sebagai berikut :

BAB I Pendahuluan

Bab ini berisi penjelasan latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian, dan sistematika laporan.

BAB II Teori Penunjang

Pada bab ini dibahas secara singkat teori-teori yang terkait dalam penulisan Tugas Akhir.

BAB III Perancangan dan Simulasi Sistem

Dalam bab ini dijelaskan mengenai perancangan dan simulasi sistem berdasarkan hasil yang didapat dari studi literatur.

BAB IV Hasil dan Analisis Data

Bab ini berisi hasil pengujian yang didapatkan baik berupa grafik maupun tabel, dianalisis dan dibahas serta berorientasi pada tujuan penelitian yang telah ditetapkan.

BAB V Penutup

Pada bab ini berisi tentang kesimpulan dari seluruh rangkaian penelitian yang telah dilakukan dan saran yang dapat dijadikan sebagai pengembangan penelitian selanjutnya.

1.7 Relevansi

Hasil yang diperoleh dari tugas akhir ini diharapkan memberikan manfaat antara lain sebagai berikut :

1. Menunjukkan referensi pengimplementasian metode *tunnel broker* dalam interkoneksi IPv6.
2. Memberikan referensi terkait parameter-parameter sistem yang harus diperhatikan dalam implementasi teknologi tersebut.

BAB 2

TEORI DASAR

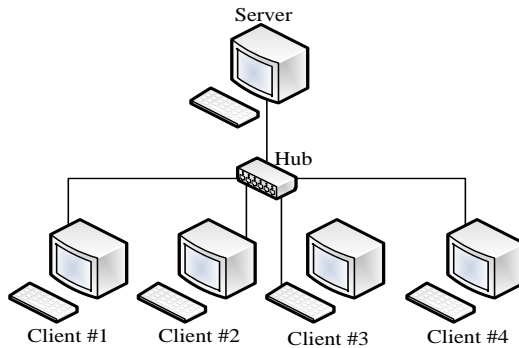
2.1 Pengenalan Jaringan

Jaringan, khususnya yang berkaitan dengan jaringan komputer mempunyai pengertian yaitu sekumpulan perangkat-perangkat yang saling terhubung sama lain saling mendukung dan berada dalam satu kesatuan sehingga antar perangkat tersebut bisa bertukar data dan dokumen. Setiap perangkat yang terhubung disebut *node*, dan setiap jaringan komputer bisa terdiri dari dua *node* bahkan lebih. Media jaringan komputer yang ada bisa terhubung melalui kabel atau nirkabel (*wireless network*). Di dalam jaringan komputer ada dua pihak yang saling berinteraksi untuk mencapai tujuan yang sama. Pihak yang menerima atau meminta layanan disebut dengan *client*, sedangkan pihak yang mengirimkan layanan disebut dengan *server*.

2.1.1 Jaringan *Local Area Network* (LAN)

Jaringan LAN atau biasa disebut dengan jaringan wilayah lokal, adalah jaringan yang mencakup wilayah kecil dan biasanya ada di wilayah perkantoran, kampus, warnet, ataupun rumah. Dasar teknologi yang digunakan dalam pemanfaatan LAN adalah IEEE 802.3 Ethernet dengan kecepatan transfer data 10, 100, atau 100 Mbit/s. Sedangkan standar untuk teknologi nirkabel menggunakan IEEE 802.11b.

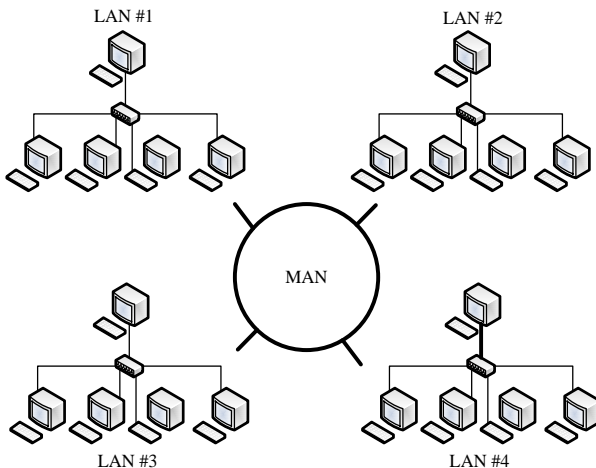
Sebuah komputer biasanya akan dijadikan *server* yang mengatur semua sistem di dalam jaringan tersebut. Dan jika *server* tersebut disambungkan ke internet maka komputer lain yang terhubung ke *server* juga akan bisa mengakses internet hanya cukup dengan satu modem dari *server*.



Gambar 2.1 Jaringan LAN

2.1.2 Jaringan *Metropolitan Area Network* (MAN)

Jaringan MAN cakupannya lebih luas dibandingkan dengan jaringan LAN, areanya mencakup sebuah negara. MAN sendiri menghubungkan jaringan-jaringan LAN ke dalam sebuah lingkungan area yang lebih besar, seperti contoh yang digunakan jaringan bank. MAN biasanya mampu menunjang data dalam bentuk teks atau suara, bahkan dapat berhubungan dengan jaringan televisi kabel atau gelombang radio.



Gambar 2.2 Jaringan MAN

2.1.3 Jaringan *Wide Area Network* (WAN)

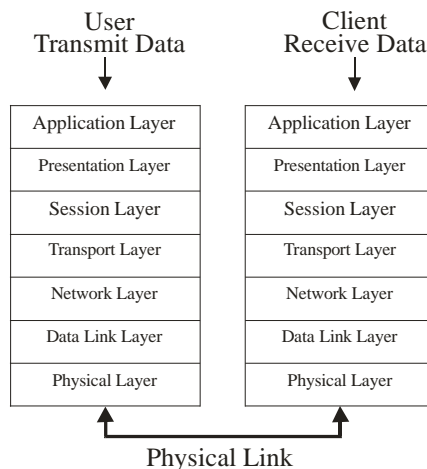
WAN diartikan sebagai jaringan yang memiliki luas yang dapat melintasi antara negara bahkan benua. WAN dimanfaatkan untuk menghubungkan jaringan lokal yang satu dengan yang lain sehingga dapat berkomunikasi. Jarak jangkauan WAN sendiri bisa mencapai 1000 KM dengan kecepatan sekitar 1.5 Mbps sampai dengan 2.4 Gbps.

2.2 Referensi Model Open System *Interconnection* (ISO)

OSI model adalah suatu referensi untuk memahami komunikasi data antara dua buah sistem yang saling terhubung. Metode lapisan digunakan didalam referensi OSI model. OSI layer membagi proses komunikasi menjadi tujuh lapisan[2]. Dari lapisan yang ada tersebut memiliki karakteristik dan fungsi masing-masing.

Peran dari OSI adalah mengidentifikasi sistem komputer dalam melaksanakan pengolahan dan transfer data. Masing-masing dari lapisan memiliki fungsi dengan tujuan agar mempermudah pelaksanaan aturan standar secara praktis. Pembagian ini juga memungkinkan adanya fleksibilitas yang berarti bila ada sebuah perubahan pada suatu lapisan, maka tidak akan mempengaruhi lapisan yang lain.

Ketujuh lapisan dalam OSI layer adalah *physical layer*, *data link layer*, *network layer*, *transport layer*, *session layer*, *presentation layer*, dan *application layer*.



Gambar 2.3 Model ISO 7 layer

2.2.1 *Physical Layer*

Physical layer atau layer fisik adalah lapisan paling pertama dari OSI layer. Lapisan ini bertanggungjawab untuk komunikasi fisik diantara perangkat. dan berperan sangat krusial dalam komunikasi data. Fungsinya adalah untuk mentransmisikan sinyal data analog maupun digital. Selain itu *physical layer* dapat digunakan untuk menentukan karakteristik dari kabel yang digunakan untuk menghubungkan komputer dalam jaringan sehingga sarana sistem pengiriman data ke perangkat lain yang terhubung dalam suatu jaringan komputer. Di lapisan inilah yang akan menjelaskan mengenai jarak terjauh yang mungkin digunakan oleh media fisik serta mengatur bagaimana cara melakukan *collision control*. Tujuan utama dari *physical layer*, yaitu:

1. Memberikan spesifikasi standar dalam berinteraksi dengan media jaringan.
2. Memberikan spesifikasi kebutuhan media untuk jaringan.
3. Menentukan karakteristik kabel untuk menghubungkan komputer dengan jaringan.
4. Mentransfer dan menentukan bagaimana bit data dikodekan.
5. Menentukan format sinyal elektrik untuk transmisi lewat media jaringan.
6. Melakukan sinkronisasi transmisi sinyal.
7. Menangani interkoneksi fisik, mekanikal, elektrik, dan prosedural.
8. Mendeteksi kesalahan selama transmisi.

2.2.2 *Data Link Layer*

Data link layer adalah lapisan kedua dari bawah dalam model OSI, yang dapat melakukan konversi *frame-frame* jaringan berisi data yang dikirimkan menjadi bit-bit mentah agar dapat diproses oleh *physical layer*. Lapisan ini merupakan lapisan yang akan melakukan transmisi data antara perangkat-perangkat yang saling berdekatan dalam sebuah *Wide Area Network (WAN)*, atau antar *node* dalam *Local Area Network (LAN)* yang sama. Beberapa perangkat yang bekerja pada lapisan ini diantaranya *Network Interface Card (NIC)*, *switch layer 2* serta *bridge* jaringan.

Fungsi spesifik dari *data link layer* adalah:

1. Penyediaan *interface* layanan bagi *network layer*.
2. Penentuan cara pengelompokan bit dari *physical layer* ke dalam *frame*.

3. Menangani *error* pada transmisi.
4. Mengatur aliran *frame*.

Layanan pentransferan data melalui fisik ditawarkan oleh *data link layer*. Pentransferan data tersebut mungkin dapat dilakukan atau tidak, beberapa protokol *data link layer* tidak mengimplementasikan fungsi *acknowledgment* untuk sebuah *frame* yang sukses diterima, dan beberapa protokol bahkan tidak memiliki pengecekan kesalahan transmisi (dengan menggunakan *checksumming*). Bila terjadi seperti itu, maka fitur *acknowledgment* dan pendeteksian kesalahan harus diterapkan pada lapisan yang lebih tinggi, seperti pada *Transmission Control Protocol (TCP)*.

2.2.3 Network Layer

Network layer bertanggung jawab dalam pemindahan data dari jaringan satu ke jaringan lain (*internetwork*). Pengalamatan *network layer* digunakan agar data bisa ditentukan tujuannya saat berpindah antar jaringan.

Fungsi secara umum *network layer* adalah:

1. Melakukan pengalamatan dan *routing* paket data.
2. Membagi aliran data biner ke paket diskrit dengan ukuran panjang tertentu
3. Mendeteksi *error*.
4. Memperbaiki *error* dengan mengirim ulang paket yang rusak
5. Mengendalikan aliran

Protokol yang tidak memiliki *network layer* hanya bisa digunakan untuk jaringan kecil. Protokol ini biasanya hanya menggunakan pengalamatan fisik (*MAC address*) dalam mengidentifikasi komputer pada jaringan. Namun akan timbul sebuah masalah bila dengan cara ini saat jaringan berkembang dalam hal jumlahnya, sehingga akan kesulitan dalam pengorganisasiannya. Contohnya saja untuk pengaturan nama dari komputer agar tidak terjadi duplikasi akan sulit diaturnya. Masalah lain adalah akan timbul *broadcast data* yang membuat boros kinerja dari jaringan.

2.2.4 Transport Layer

Layanan yang ada pada *transport layer* mencakup transportasi data dari ujung ke ujung dan dapat memuat koneksi antara host pengirim dan host penerima. *Transport layer* memiliki fungsi dalam

penyediaan mekanisme pengiriman atau penerimaan dari berbagai jenis data pada saat bersamaan melalui satu media *network* yang biasa disebut teknik *multiplexing*, metode aplikasi *upper layer*, membuat *session* dan memutuskan koneksi yang terbentuk antara dua buah host di jaringan, setelah melalui mekanisme *three-way handshake*. Protokol yang berlaku pada *transport layer* diantaranya *UDP (User Datagram Protocol)* dan *TCP (Transmission Control Protocol)*.

2.2.5 Session Layer

Session layer bertanggung jawab dalam pembentukan, pengelolaan dan memutuskan *session* antar *presentation layer*. Disini juga menyediakan kontrol dialog antar *node*. *Session layer* akan melakukan koordinasi komunikasi antar sistem dan mengorganisasi komunikasinya dengan menawarkan mode komunikasi satu arah (*simplex*), komunikasi dua arah bergantian (*half duplex*) dan komunikasi dua arah (*full duplex*). Pada intinya *session layer* menjaga terpisahnya data dari aplikasi satu dengan yang lainnya.

2.2.6 Presentation Layer

Pada lapisan ini hanya terdapat satu fungsi, yaitu translasi berbagai macam tipe pada *syntax* sistem. Contohnya adalah koneksi antar PC dan *mainframe* membutuhkan konversi *Extended Binary Coded Decimal Interchange Code (EBCDIC) character encoding format* ke *ASCII* dan banyak faktor yang harus dipertimbangkan. Data akan dikompresi oleh layer ini. Layer ini pada dasarnya sebagai penerjemah, pengkodean dan pengkonversi.

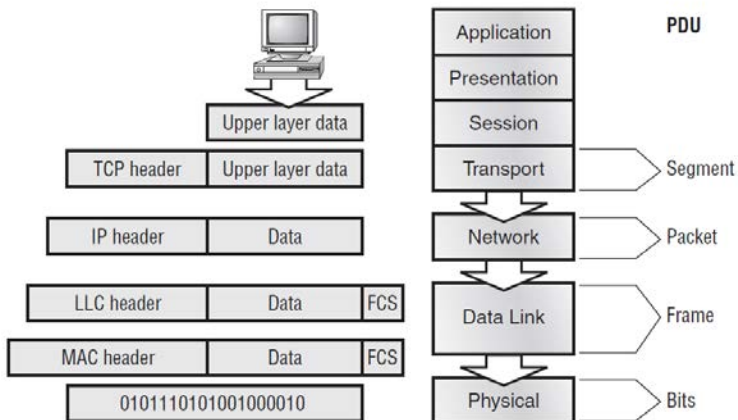
2.2.7 Application Layer

Lapisan ketujuh dalam OSI model ini adalah lapisan yang menyediakan *interface* antara aplikasi yang digunakan untuk berkomunikasi dan jaringan yang mendasarinya di mana pesan akan dikirim. *Application layer* menggunakan protokol yang diimplementasikan dalam aplikasi dan layanan. Layer ini berbeda dengan layer yang lain karena tidak memberi layanan pada layer lain, namun akan memberikan layanan pada aplikasi di luar OSI model.

Application layer berfungsi sebagai *interface* antara aplikasi antara aplikasi yang dihadapi *user undersource* jaringan yang diakses. Atau singkatnya lapisan ini menjembatani interaksi manusia dengan perangkat lunak atau *software* aplikasi.

2.3 Enkapsulasi

Jika sebuah host melakukan pengiriman data melewati jaringan yang ada ke perangkat lainnya, data tersebut melalui sebuah proses yang bernama enkapsulasi dan akan dibungkus dengan informasi protocol dalam tiap OSI layer. Lapisan tersebut akan melakukan komunikasi dengan lapisan yang sama pada penerima. Pada gambar 2.4 dijelaskan tentang proses enkapsulasi. Dalam melakukan komunikasi, tiap-tiap lapisan menggunakan *Protocol Data Unit* (PDU) yang memiliki lampiran informasi control untuk data pada tiap lapisan model. PDU tersebut akan menempel pada data dengan cara mengenkapsulasikannya pada lapisan model OSI dan PDU memiliki nama yang berbeda berdasarkan informasi yang disediakan dalam tiap *header* PDU hanya akan bisa dibaca oleh lapisan yang sejenis pada perangkat pertama.



Gambar 2.4 Proses enkapsulasi

Seperti pada gambar 2.4, ditunjukkan PDU pada setiap lapisannya. *Data stream* dan *upper layer data* diturunkan ke *transport layer*. *Data stream* tersebut dipecah menjadi bagian-bagian kecil dan diberikan *header transport layer* disetiap *data field* nya yang dinamakan *segment*. Setiap *segment* memiliki urutan tertentu yang nantinya dapat disusun kembali pada sisi penerima sesuai dengan urutannya ketika seperti ketika dikirimkan.

Segmen-segmen yang ada akan diturunkan ke *network layer* untuk dilakukan pengalamatan jaringan dan routing. Pengalamatan *logic*,

contohnya IP digunakan agar segmen dapat mencapai tujuannya dengan benar. Pada *network layer* akan menambahkan *header control* pada segmen. PDU pada bagian ini dinamakan *packet* atau *datagram*.

Setelah melewati *network layer*, maka data akan melewati *data link layer*. Data sebelumnya yang berbentuk paket, pada lapisan ini akan dilakukan enkapsulasi sehingga menjadi *frame*. *Header* yang ada pada *frame* membawa informasi mengenai alamat perangkat dari pengirim dan penerima. Jika perangkat tersebut berada pada jaringan yang lain, maka *frame* itu akan dikirimkan ke router untuk diroutingkan melalui internetwork. Setelah *frame* sampai pada tujuan, *frame* yang baru digunakan agar paket dapat sampai ke host tujuan.

Kemudian *frame* tersebut akan sampai pada *physical layer*. Pada lapisan ini, *frame* akan direpresentasikan dalam bentuk sinyal digital agar bisa diteruskan dalam jaringan. Lapisan ini akan melakukan encoding digit-digit *frame* menjadi sinyal digital yang kemudian akan dibaca pada perangkat lokal. Perangkat penerima akan melakukan sinkronisasi dan decoding terhadap sinyal digital yang diterima. Perangkat penerima akan melakukan pembangunan *frame*, mengecek kesalahan atau disebut *Cyclic Redundancy Check*, dan membandingkan hasilnya dengan bagian dari *Frame Check Sequence*. Jika cocok, maka paket tersebut akan diambil dari *frame* dan *frame* sisa akan dibuang atau disebut dengan proses dekapsulasi. Paket akan diberikan pada *network layer* dan alamatnya akan dicek. Jika sudah sesuai, maka segmen akan ditarik dari paket dan paket sisa akan dibuang. Segmen akan diproses pada *transport layer* dan dibangun ulang *data stream* dan diteruskan ke aplikasi *upper layer*.

2.4 File Transfer Protocol (FTP)

Protokol file transfer adalah protokol yang mengatur tentang mekanisme pertukaran file antar komputer dalam jaringan yang mendukung protokol TCP/IP, seperti pada internet. Dalam FTP akan dipastikan bahwa file akan diterima tanpa terjadinya *loss* pada file tersebut. FTP menggunakan protokol TCP pada tataran *transport layer*. Tujuan dari adanya FTP adalah melakukan berbagi file antar komputer, melakukan secara langsung ataupun implisit mengenai penggunaan komputer *remote*, melindungi pengguna dari variasi dalam sistem penyimpanan file antar host, dan adanya pertukaran data yang andal dan efisien[3].

FTP bekerja berdasarkan konsep *client/server*. FTP *client* adalah *client* yang melakukan permintaan koneksi kepada FTP *server* untuk melakukan pertukaran file. Sedangkan FTP *server* adalah penyedia file yang akan diminta oleh *client*, dalam pertukaran *data server* akan meminta otentifikasi data berupa *user name* dan *password* agar *client* bisa mengakses file yang diinginkan. Namun kelemahan dari FTP ini hanya mampu digunakan dalam transfer data, tidak bisa membuka dokumen seperti pada *windows explorer*. Jikapun bisa dibuka, itu hanya sebatas dalam format *editor text*. Pada transfer data akan digunakan dua buah mode yaitu binary dan mode ASCII. Mode binary ditujukan untuk data 8 bit sedangkan mode ASCII untuk data 7 bit.

2.5 IPv6

Penggunaan jaringan IPv4 dalam pengalamatan jaringan sejak tahun 1970 dengan ketersediaan alamat hingga 2^{32} atau sekitar 4.294×10^9 sebenarnya amatlah mudah diimplementasikan dan dioperasikan hingga masa sekarang. Namun desain IPv4 ternyata tidak mampu mengatasi dampak perkembangan jaringan internet yang semakin pesat, diantaranya:

1. Dibutuhkan jumlah ketersediaan alamat dalam mendukung perkembangan internet yang pesat.
2. Perlunya sebuah kemampuan dari *router backbone* internet untuk pengelolaan *tabel routing* yang besar.
3. Kebutuhan keamanan yang lebih kuat.
4. Kebutuhan akan *Quality of Service (QoS)* yang lebih baik dalam pengiriman *real time*.

Berdasarkan masalah itulah dikembangkan *Internet Protocol Next Generation (IPng)* atau lebih umum disebut IPv6. Namun pengimplementasian IPv6 terhadap jaringan yang ada, haruslah melalui metode transisi yang tepat agar tidak mengganggu jaringan IPv4 yang sudah ada.

2.4.1 Perubahan dari IPv4 ke IPv6

Perbedaan dari IPv4 dan IPv6 dapat dikategorikan dalam beberapa hal, berikut ulasanya:

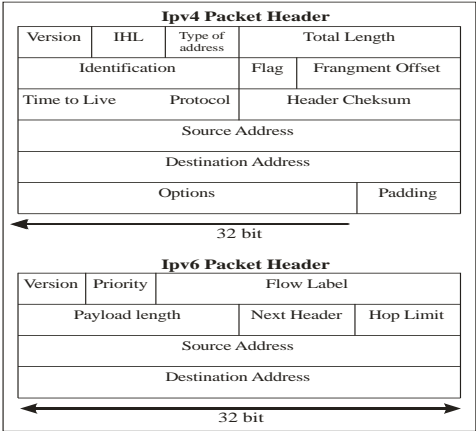
2.4.1.1 Kapasitas Perluasan Alamat

Peningkatan perluasan ukuran dan kapasitas alamat IPv6 terhadap IPv4 terlihat pada jumlah bit dari 32 bit menjadi 128 bit. Hal

ini disebabkan untuk mendukung peningkatan hirarki atau kelompok pengalaman, peningkatan jumlah atau alamat dialokasikan pada *node* yang akan mempermudah *node* sehingga bisa dilakukan konfigurasi alamat secara otomatis.

2.4.1.2 Penyerderhanaan Format Header

Beberapa kolom pada *header* IPv4 telah dihilangkan atau dapat dibuat sebagai *header* pilihan. Hal ini mengurangi biaya pemrosesan hal-hal umum pada penanganan paket IPv6 dan membatasi *bandwidth* pada *header* IPv6. Dengan demikian pada prosesnya akan lebih efisien.



Gambar 2.5 Perbandingan format header IPv4 dan IPv6

Tabel 2.1 Keterangan header IPv6

<i>Version</i>	4-bit nomor versi Internet Protocol 6
<i>Traffic Class</i>	8-bit <i>fieldtraffic class</i>
<i>Flow Label</i>	20-bit <i>flow label</i>
<i>Payload Length</i>	16-bit <i>unsigned</i> integer. Panjang <i>payload</i> IPv6 dalam oktet
<i>Next Header</i>	8-bit <i>selector</i> . Mengindetifikasi tipe <i>header</i> yang langsung mengikuti <i>header</i> IPv6. Menggunakan nilai yang sama seperti <i>field protocol</i> Iv4.
<i>Hop Limit</i>	8-bit <i>unsigned</i> integer. Dikurangi dengan 1 oleh setiap <i>node</i> yang meneruskan paket
<i>Source Address</i>	128-bit alamat asal dari paket

<i>Destination Address</i>	128-bit alamat penerima yang dituju dari paket (tidak selalu penerima terakhir jika ada <i>header routing</i>)
----------------------------	---

2.4.1.3 Option dan Extension Header

Perubahan yang terjadi pada *header-header* IP yaitu dengan adanya pengkodean *header options* pada IP dimasukkan supaya lebih efisien dalam *packet forwarding*, agar tidak terlalu rumit dalam pembatasan panjang *header options* yang ada dalam paket IPv6 dan sangat fleksibel untuk mengenalkan *header options* yang baru nantinya.

2.4.1.4 Kemampuan Pelabelan Aliran Paket

Fitur terbaru yang dimasukkan dalam IPv6 adalah kemampuan pelabelan paket atau pengklasifikasian paket yang memerlukan penanganan khusus, seperti *Quality of Service* dan *real time*.

2.4.1.5 Kemampuan Autentifikasi dan Privasi

Adanya kemampuan tambahan untuk mendukung otentifikasi, integritas data dan data yang dapat dikategorikan penting juga akan dispesifikasikan dalam alamat IPv6. Salah satu dampak dari perluasan alamat IPv6 adalah ruang *address* yang kontinyu dengan menghilangkan konsep kelas.

Untuk memudahkan perbandingan IPv4 dan IPv6 disajikan melalui tabel 2.2 sebagai berikut:

Tabel 2.2 Perbandingan IPv4 dan IPv6

IPv4	IPv6
Panjang alamat 32 bit (4 bytes)	Panjang alamat 128 bit (16 bytes)
Dukungan terhadap <i>IPsec optional</i>	Dukungan terhadap <i>IPsec</i> dibutuhkan
Konfigurasi secara manual	Tidak perlu konfigurasi manual, bisa menggunakan <i>address autoconfiguration</i>
Fragmentasi dilakukan oleh pengirim dan menurunkan kinerja router	Fragmentasi hanya dilakukan pengirim

<i>Checksum</i> termasuk pada <i>header</i>	<i>Checksum</i> tidak masuk di <i>header</i>
<i>Header</i> mengandung <i>option</i>	<i>Data optional</i> dimasukkan semua ke dalam <i>extension header</i>
Tidak mensyaratkan ukuran paket <i>link-layer</i> dan harus bisa menyusun kembali paket berukuran 576 byte	Paket <i>link-layer</i> harus mendukung ukuran paket 1280 byte dan harus bisa menyusun kembali paket sebesar 1500 byte
Menggunakan <i>ARP Request</i> secara <i>broadcast</i> untuk menterjemahkan IPv4 ke alamat <i>link-layer</i>	<i>ARP Request</i> telah digantikan oleh <i>Neighbor Solicitation</i> secara <i>multicast</i>
Untuk mengelola di keanggotaan grup pada subnet lokal digunakan <i>Internet Group Managemet Protocol (IGMP)</i>	IGMP digantikan oleh <i>Multicast Listener Discovery (MLD)</i>

2.4.2 Pengalamatan IPv6

Ada tiga tipe pengalamatan yang berbeda dalam IPv6, dengan penjelasan sebagai berikut:

2.4.2.1 Unicast Address (*one-to-one*)

Unicast address adalah jenis IP address yang hanya mengidentifikasi sebuah *interface*. Paket yang dikirim ke *unicast address* hanya akan diterima oleh *interface* pengguna alamat tersebut. Pada alamat unicast dibagi menjadi 3 bagian, yaitu : alamat *link local* yang digunakan dalam satu link jaringan, alamat *site local* yang sama dengan alamat *private* pada IPv4, dan alamat global, yaitu alamat *public* yang digunakan oleh *Internet Service Provider*.

2.4.2.2 Multicat Address (*one-to-many*)

Multicast address adalah alamat yang digunakan untuk berkomunikasi beberapa *interface* (biasanya dalam *node* yang berbeda) dengan menunjuk host dari grup atau bisa dikatakan mampu mengidentifikasi sekumpulan *interface*. Paket yang dikirimkan ke

multicast address akan diterima oleh semua *interface* yang menggunakan alamat tersebut. Alamat ini di desain untuk menggantikan alamat *broadcast* pada IPv4.

2.4.2.3 Anycast Address

Anycast address menunjuk host dari grup, tapi paket yang dikirim hanya berasal dari satu host saja. Pada alamat jenis ini, sebuah *address* diberikan pada beberapa host, untuk mendefinisikan kumpulan *node*. Jika ada paket yang dikirim ke alamat ini, maka router akan mengirimkan paket ke host terdekat yang memiliki *anycast address* sama. Atau sederhananya pemilik paket akan menyerahkan kepada router tujuan yang paling cocok dalam pengiriman paket tersebut. Pemakaian *anycast address* ini misalnya terhadap beberapa *server* yang memberikan layanan seperti DNS. Beban terhadap *server* akan terdistribusi secara merata.

2.4.3 Representasi Alamat IPv6

Penulisan alamat pada IPv6 berbeda dengan penulisan dalam IPv4. Jika pada IPv4 ditulis dalam bentuk desimal yang terbagi menjadi 4 bagian, maka pada alamat IPv6 ditulis dalam heksadesimal yang terbagi menjadi delapan bagian. Format penulisan alamat IPv6 adalah x:x:x:x:x:x:x, “x” adalah empat digit bilangan heksadesimal. Contohnya adalah :

FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

Jika nilai “x” bernilai “0” maka dapat disederhanakan menjadi “::”. Contohnya sebagai berikut:

1080:0:0:0:8:800:200C:417A	(unicast address)
FF01:0:0:0:0:0:0:101	(multicast address)
0:0:0:0:0:0:0:1	(loopback address)
0:0:0:0:0:0:0:0	(unspecified address)

Dapat direpresentasikan menjadi :

1080::8:800:200C:417A	(unicast address)
FF01::101	(multicast address)
::1	(loopback address)

::

(*unspecified address*)

Model x:x:x:x:x:d:d:d:d dimana “d:d:d:d” adalah alamat IP semacam 192.205.25.6 yang digunakan dalam *tunneling* otomatis. Contohnya adalah:

0:0:0:0:0:192:205:25:6 atau **::192.205.25.6**

0:0:0:0:0:FFFF:192.205.25.7 atau **:FFFF:192.205.25.7**

Jadi jika sekarang mengakses sebuah alamat di internet dengan format **192.205.25.6** maka kedepannya akan digantikan dengan format seperti **::BA66:070:18**.

Prefix pada alamat IPv6 sama seperti pada IPv4 yang ditulis dalam notasi CIDR. Penulisan alamat pada format IPv6 adalah alamat_IPv6/panjang_prefix. Panjang *prefix* adalah bilangan decimal yang menyatakan berapa banyak jumlah bit yang diambil dari sebelah kiri untuk digunakan sebagai *prefix*. Contohnya sebagai berikut :

3FFE:CD30:0:0:FE34:0:0/60

Angka 60 bit awal menunjukkan bagian *network* bit. Jika pada IPv4 mengenal pembagian kelas A, B, dan C maka pada IPv6 pun dilakukan pembagian kelas berdasarkan format *prefix*, yaitu format bit awal alamat. Misalnya

3FFE:CD30:0:0:FE34:0:0/60

Jika diperhatikan 4 bit awal yaitu hexa “3” yang didapatkan *prefix*nya untuk 4 bit awal adalah 0011 (nilai biner dari 3).

2.6 Mekanisme Transisi IPv6

Sebuah kelompok penelitian IETF NGtrans (*Next Generation Transition*) telah merancang mekanisme transisi IPv4-IPv6 untuk mengatasi berbagai kebutuhan jaringan yang berbeda[4]. Karena pada dasarnya IPv4 dan IPv6 tidak kompatibel satu sama lain. Mekanisme transisi tersebut mempunyai dua tujuan yaitu :

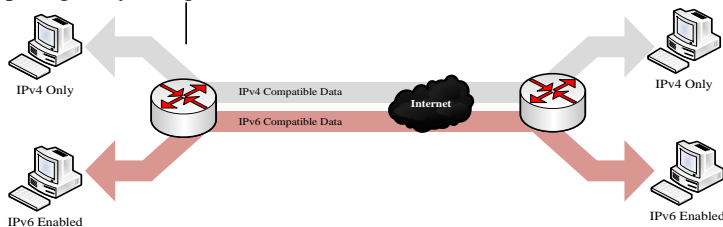
1. Membuat agar terminal dari IPv6 dapat berkomunikasi dengan terminal IPv4.

2. Melewatkan paket IPv6 melalui jaringan IPv4 yang sudah tersedia.

Beberapa mekanisme transisi IPv4 dan IPv6 yang digunakan adalah *dual stack*, *tunneling* dan translasi.

2.5.1 Dual Stack

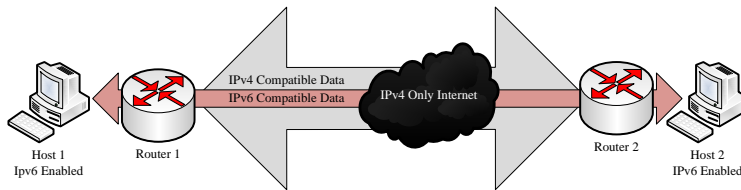
Mekanisme *dual stack* mencakup dua buah protokol yang beroperasi bertumpuk secara paralel sehingga memungkinkan *node* jaringan untuk beroperasi baik melalui protokol IPv4 atau IPv6[5]. Metode ini diimplementasikan dalam *end system* nya dan juga *node* jaringannya. Panduan yang mengatur tentang mekanisme *dual stack* ini adalah RFC 2983. Di dalamnya dijelaskan bahwa *node* jaringan IPv4 dan IPv6 akan ditumpuk. Aplikasi IPv4 menggunakan IPv4 *stack* dan IPv6 menggunakan IPv6 *stack*. Pengiriman data diputuskan berdasarkan versi *header* IP untuk penerima dan tujuan jenis alamat pengiriman. Jenis alamat didapat dari pencarian DNS. Mekanisme *dual stack* saat ini paling banyak digunakan dalam transisi IPv4 dan IPv6.



Gambar 2.6 Metode *dual stack*

2.5.2 Tunneling

Dua buah jaringan yang memiliki perbedaan agar dapat terhubung harus menggunakan penanganan khusus, namun dalam penerapannya secara nyata sebenarnya sangatlah sulit. *Tunneling* dapat dimisalkan ada dua buah host, satu berperan sebagai host sumber dan yang lain berlaku sebagai host tujuan dari jaringan yang memiliki jenis sama, akan tetapi jaringan yang terletak diantara keduanya berbeda jenis.



Gambar 2.7 Metode *tunneling*

Permasalahan dengan 2 jaringan yang berbeda dapat dipecahkan dengan mekanisme *tunneling*. Dalam pengiriman paket IP ke host2, host1 akan membuat paket yang berisi alamat IP host2, menyisipkannya ke *frame ethernet* yang dialamatkan ke router 1 dan meneruskannya melalui ethernet, proses ini dinamakan enkapsulasi. Saat router menerima *frame*, router tersebut akan menghapus paket IP dan menyisipkannya ke *field payload*. *Network layer WAN* kemudian mengalamatkannya ke router WAN tujuan. Sesaat paket sampai, router akan menghapus IP dan mengirimkannya ke host2 pada *frame ethernet*[6], proses ini dinamakan dekapsulasi.

Dalam *tunneling* masih terbagi lagi menjadi beberapa macam cara, diantaranya :

1. Mekanisme 6over4

Paket IPv6 dapat secara otomatis dienkapsulasi melalui jaringan IPv4 dengan menggunakan IP *multicast*. 6over4 memberikan penyelesaian masalah konektivitas *node* IPv6 yang tersebar di seluruh domain IPv4 tanpa konektivitas IPv6 secara langsung. Mekanisme ini memungkinkan *node*, pada *physical link*, yang secara langsung terhubung router IPv6 menjadi *node* IPv6 yang berfungsi secara penuh.

2. Mekanisme 6to4

Pada 6to4 memungkinkan domain IPv6 terisolasi untuk dihubungkan melalui IPv4 jaringan dan *remote* jaringan IPv6. Hal tersebut membuat infrastruktur dari IPv4 sebagai *link non-broadcast virtual*, sehingga alamat IPv4 tertanam dalam alamat IPv6 yang digunakan mencari tujuan pengiriman. Alamat IPv4 yang tertanam dapat dengan mudah diekstrak dan seluruh paket IPv6

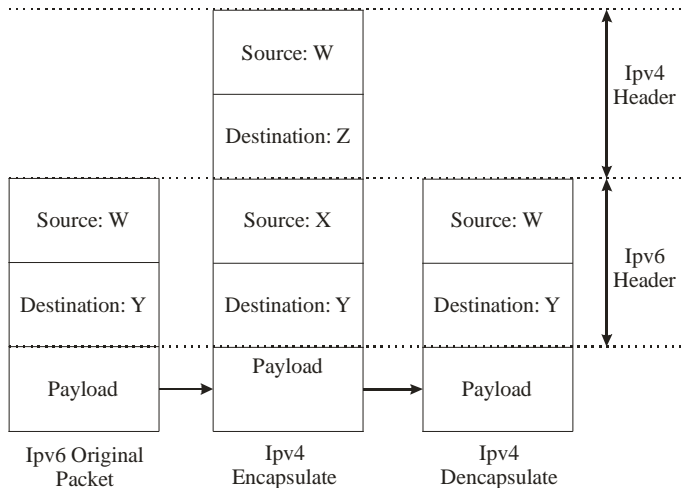
disampaikan melalui jaringan IPv4, dikemas dalam sebuah paket IPv4.

3. IPv6 Tunnel Broker

Tunnel broker digunakan sebagai aktifasi *tunnel* dan proses registrasi dari *user* IPv4. Tugas pokoknya adalah mengatur dari pembentukan, perubahan, dan penghapusan *tunnel* sesuai dengan keinginan *user*. Pada pengaplikasiannya *tunnel broker* akan membagi beban jaringan kepada *tunnel server* yang bersangkutan saat pembentukan *tunnel* tersebut dibentuk, dimodifikasi ataupun dihapus. Pendaftaran alamat IPv6 dari user kemudian memasukkannya ke dalam *DNS server* juga dilakukan oleh *tunnel broker*.

Tunnel broker harus bisa mendukung IPv4 tetapi tidak harus bisa mendukung IPv6. Ini dikarenakan *tunnel broker* hanya berhubungan secara langsung dengan IPv4 dan hubungan *tunnel broker* dan *tunnel server* dapat berupa IPv6 ataupun IPv4.

Mekanisme *tunneling* dilakukan dengan cara enkapsulasi paket IPv6 dengan *header* IPv4, kemudian paket akan dikirimkan kepada jaringan IPv4. Enkapsulasi akan dilakukan oleh pengirim dan pada penerima akan melakukan proses sebaliknya yaitu de-enkapsulasi. Enkapsulasi adalah memberikan *header* IPv4 pada paket IPv6 agar paket dapat diroutingkan atau dilewatkan pada jaringan IPv4 namun akan ada penambahan *header* IPv4, maka paket akan bertambah besar sesuai panjang *header* IPv4 yaitu 20 byte. Sedangkan proses dekapsulasi adalah menghilangkan *header* IPv4 yang melekat pada paket IPv6 agar paket tersebut bisa diterima oleh perangkat yang menggunakan IPv6.



Gambar 2.8 Proses enkapsulasi paket mekanisme *tunneling*

2.5.3 Translasi

Metode ini kurang umum digunakan dalam transisi IPv4 dan IPv6. Hal ini dikarenakan dalam metode ini dibutuhkan perangkat tambahan untuk mentranslasi paket IPv4 ke paket IPv6 atau sebaliknya. Pada intinya, dalam mekanisme ini dilakukan cara menerjemahkan protokol IPv4 ke IPv6. Beberapa metode translasi yang ada yaitu :

1. *ALG (Application Level Gateway)*

Mekanisme host IPv6 hanya berkomunikasi dengan IPv4 melalui sebuah *Dual Stack Proxy*.

2. *NAT-PT (Network Address Translator Protocol Translator)*

Metode ini memungkinkan host dan aplikasi *native* IPv6 untuk berkomunikasi dengan host dan aplikasi IPv4. Tiap-tiap host yang berperan sebagai *address translator* menyimpan sekumpulan alamat yang diberikan secara dinamis ke host IPv6 dan sebuah sesi akan dibentuk antara dua host yang mendukung protokol yang berbeda. NAT-PT memberikan dukungan translasi *header* dan alamat. Mekanisme ini tidak mendukung implementasi sekuriti *end-to-end* dan memerlukan ruang IPv4 yang besar. Merujuk

dalam tabel translasi dimana alamat IP dari *nodehost* IPv6 dan *pool address* pada translator bersesuaian, translasi sebuah alamat IP dan bagian *header* IP diubah untuk IPv4 dan IPv6.

Dalam mempersiapkan *pool address* untuk koneksi yang diinisiasi ke arah IPv4 dan IPv6, dimungkinkan untuk menggunakan *Network Address Port Translation (NAT-PT)* yang membagi sebuah alamat ke dua atau lebih *nodehost* IPv6 dengan mengganti nomor *port* untuk setiap koneksi TCP atau UDP. Ketika sebuah *nodehost* mengirimkan data bervolume besar ke *nodehost* yang lain, data dikirimkan dalam bentuk IP. Untuk paket-paket IP ini, data seharusnya tidak difragmentasi ketika dikirimkan dari *node* sumber ke *node* tujuan. Walaupun perbedaan panjang header IP dari kedua protokol melebihi *Maximum Transmission Unit (MTU)* dari translator dikarenakan *link* pada perbatasan IPv4 dan IPv6.



Gambar 2.9 Metode NAT-PT

3. BIS (*Bump In Stock*)

Yaitu sebuah mekanisme yang membolehkan aplikasi IPv4 berkomunikasi dengan host IPv6.

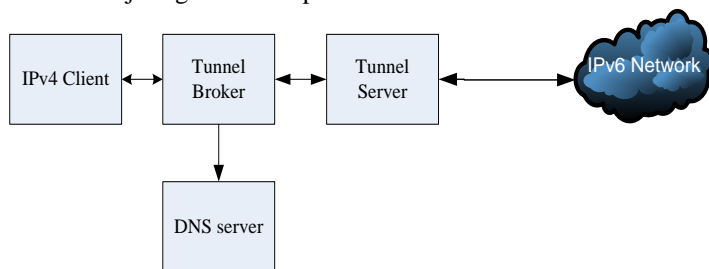
4. SOCK Gateway

Mekanisme translasi yang menerima koneksi *enchanced SOCK* dan meneruskannya ke jaringan IPv4 atau IPv6.

2.7 IPv6 Tunnel Broker

IPv6 tunnel broker adalah salah satu mekanisme transisi IPv4 ke IPv6 yang cukup mudah dalam penggunaannya. Metode tersebut akan menyediakan konfigurasi untuk melakukan *tunneling* IPv6 melalui IPv4 kepada *user* IPv4 yang terhubung dalam jaringan internet. Sederhananya, *IPv6 tunnel broker* seperti *Internet Service Provider (ISP)* dengan IPv6 yang menyediakan koneksi IPv6 untuk *user* yang

telah terhubung ke jaringan internet lewat IPv4. Berdasarkan penjelasan yang ada, *tunnel broker* bisa menjadi alternatif yang lebih unggul dalam transisi IPv4 ke IPv6 pada kasus *user* yang ingin mengakses sebuah situs ataupun server dengan pengalamatan IPv6 namun harus melewati jaringan IPv4 yang berada diantara *user* dan server IPv6 tersebut. Bila penggunaan mekanisme dual stack dan translasi pada kasus tersebut tidak bisa digunakan karena router ataupun perangkat *endpoint* yang ada, dimiliki oleh ISP sehingga tidak memungkinkan untuk dilakukan konfigurasi. Namun pada metode *tunnel broker*, *user* bisa melakukan konfigurasi dari perangkat *user* itu sendiri yang ada untuk membentuk kanal melalui jaringan IPv4 kepada server IPv6.



Gambar 2.10 Arsitektur IPv6 *tunnel broker*

2.6.1 Tunnel Broker

Tunnel broker adalah wadah koneksi dari *user* pada jaringan IPv4 untuk melakukan proses pendaftaran dan pengaktifan *tunnel*. Fungsinya adalah untuk mengatur pembentukan, modifikasi, dan penghapusan *tunnel* sesuai dengan permintaan dari *user*. Dalam prakteknya, *tunnel broker* bertanggung jawab untuk membagi beban jaringan kepada *tunnel server*, caranya adalah dengan mengirimkan konfigurasi kepada *tunnel server* yang bersangkutan pada saat *tunnel* tersebut dibentuk, dimodifikasi, atau dibubarkan. *Tunnel broker* juga harus mendaftarkan alamat IPv6 *user* dan memasukkannya ke dalam *DNS server*.

Tunnel broker haruslah mendukung IPv4 namun tidak harus mendukung IPv6, karena *tunnel broker* akan berhubungan secara langsung dengan internet melalui jaringan IPv4 dan hubungan antara *tunnel broker* dan *tunnel server* dapat berupa IPv6 maupun IPv4. Selain

itu *tunnel broker* dapat dilengkapi dengan otentifikasi, otorisasi dan akuntansi untuk manajemen *user* dan akuntansi *tunnel*.

2.6.2 Tunnel Server

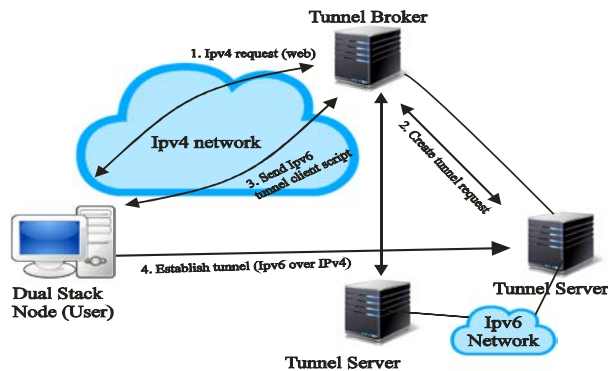
Tunnel server adalah router *dualstack* (IPv4 dan IPv6) yang terhubung dengan jaringan IPv6. *Tunnel server* mempunyai tugas menerima seluruh konfigurasi yang dikirim oleh *tunnel broker* pada saat pembangunan, modifikasi dan penghapusan *tunnel* pada sisi *server*.

2.6.3 Domain Name Service (DNS) Server

DNS server bertugas untuk menerjemahkan dari nama domain ke alamat IP atau sebaliknya dari pemakai yang telah membentuk *tunnel*. *Server* ini harus mendukung IPv6, karena domain yang digunakan merupakan jaringan IPv6. Namun penggunaan *DNS server* bersifat opsional.

2.6.4 Mekanisme Tunnel Broker

Mekanisme kerja dari *tunnel broker* dapat dilihat pada ilustrasi berikut:



Gambar 2.11 Mekanisme *tunnel broker*

1. *User* akan menghubungi *tunnel broker* dan akan dilanjutkan proses pendaftaran (biasanya mengisi *form* pada *web*) dan *user* akan diberikan hak untuk mengakses layanan *tunnel*.
2. *User* menghubungi kembali *tunnel broker* dan setelah proses otentifikasi *user* tersebut memberikan informasi tentang

konfigurasi dari host baik berupa alamat IP, *operating system* dan perangkat lunak pendukung IPv6.

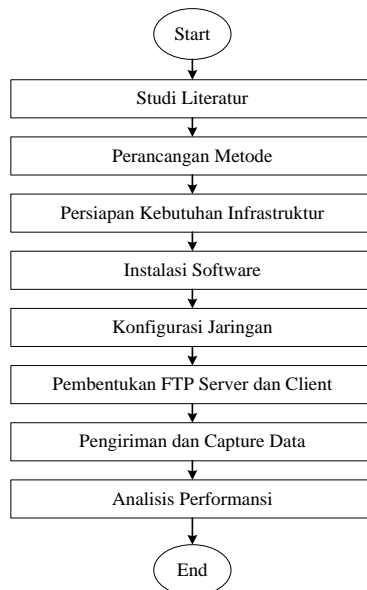
3. *Tunnel broker* akan mengkonfigurasi *tunnel* di sisi jaringan (*tunnel server*) dan *DNS server*.
4. Skrip aktivasi *tunnel* akan diberikan di sisi *user*. Jika proses ini berhasil, maka *user* artinya telah terhubung ke jaringan IPv6 melalui *tunnel server* yang telah ditentukan *tunnel broker*.
5. *User* dapat meminta modifikasi ataupun penghapusan *tunnel* dengan cara mengakses *tunnel broker* lagi.

BAB 3

PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dibahas tentang sistem dan metode pengujian sistem. Tujuan yang ingin dicapai adalah mendapatkan hasil unjuk kerja dari pengiriman paket melalui aplikasi FTP pada jaringan *tunnel broker* yang dilakukan oleh *FTP client* kepada *FTP server*. Hasil yang didapatkan akan dibandingkan pada pengiriman paket aplikasi FTP berdasarkan tipe file dan ukurannya dengan jaringan IPv4 dan IPv6 murni.

Selain itu juga akan dibahas tentang persiapan perancangan sistem sebelum dilakukan implementasi dan tahapan dalam pengambilan data sesuai dengan parameter yang telah ditentukan. Alur dari proses pengerjaan tugas akhir dapat dilihat pada gambar berikut:



Gambar 3.1 Diagram alir perancangan dan implementasi sistem

Dari gambar 3.1 diperoleh sistematika tahapan pengerjaan tugas akhir. Pertama-tama dilakukan studi literatur terkait teori penunjang

mekanisme pembentukan *IPv6 tunnel broker* dan juga aplikasi FTP. Setelah itu dirancanglah metode yang tepat dalam pengimplementasian sistem beserta parameter-parameter yang dibutuhkan untuk analisa perbandingan performa transfer data yang berjalan pada aplikasi FTP di jaringan *tunnel broker*, IPv4 dan IPv6. Kemudian dicarilah perangkat pendukung infrastruktur baik berupa perangkat keras maupun perangkat lunak yang mendukung sistem. Tahap berikutnya ialah melakukan instalasi perangkat lunak sesuai dengan kebutuhan yang ada. Bila sudah melakukan instalasi perangkat lunak, lalu mulai konfigurasi jaringan. Untuk yang pertama dilakukan adalah konfigurasi jaringan *tunnel broker* dan dilakukan pengambilan data. Ketika jaringan sudah terbentuk, dilanjutkan dengan pembuatan *server* dan *client* yang berjalan pada aplikasi FTP. Hal ini dikarenakan proses analisis data yang dibutuhkan adalah proses transfer data melalui protokol FTP, sehingga harus dibangun secara khusus *server* dan *client* yang berbasis FTP.

Dari hasil pengiriman data, akan diperoleh nilai *latency* dan *throughput* dari jaringan *tunnel broker*. Selanjutnya dari data tersebut dibandingkan satu sama lain antar jaringan yang ada. Terakhir dilakukan proses analisis data dan penarikan kesimpulan dari tugas akhir ini.

3.1 Parameter Simulasi

Dalam analisis unjuk kerja interkoneksi IPv6 dan IPv4 dengan metode *IPv6 tunnel broker* ini parameter simulasi yang digunakan adalah sebagai berikut:

- Metode yang digunakan dalam jaringan adalah *IPv6 tunnel broker*, IPv4 dan IPv6.
- Tipe file yang digunakan ada lima buah yaitu, txt, rar, pdf, mp3, iso untuk pengukuran berdasarkan beda ukuran file dan dua tipe file untuk perbandingan mode pengiriman yaitu txt dan pdf.
- Masing-masing tipe akan dilakukan pengambilan data sebanyak 10 kali, jadi akan ada 210 kali pengambilan data dari total keseluruhan.
- Parameter pertama yang diamati dan dianalisis dalam pengambilan data adalah *latency*. *Latency* adalah waktu yang dibutuhkan dalam pengiriman paket dari *server* ke *client* dan dinyatakan dalam satuan *second*.

- Parameter uji kedua adalah *throughput*. *Throughput* dapat diartikan sebagai kecepatan rata-rata sebuah pengiriman paket tiap detiknya dari *server* ke *client*.

3.2 Perangkat Pendukung

Dalam tugas akhir ini, diperlukan adanya perangkat pendukung baik berupa *hardware* maupun *software*. Kedua jenis perangkat tersebut dibutuhkan dalam implementasi dan pengambilan data dan saling mendukung satu sama lainnya.

3.2.1 Hardware (Perangkat Keras)

Ada 3 jenis perangkat keras yang digunakan dalam tugas akhir ini, yaitu *personal computer server*, *personal computer client*, dan juga *personal computer router*. Semua perangkat tersebut akan menjadi sebuah sistem yang saling berkaitan sesuai dengan fungsinya masing-masing dengan penjelasan sebagai berikut:

3.2.1.1 Personal Computer Server

Personal computer server akan melakukan tugasnya sebagai sumber atau *server* yang menerima layanan permintaan FTP dari *FTP client*. Komputer *server* akan direpresentasikan dengan nama *host1*. Spesifikasi dari *PC server* adalah:

- Seri : Vaio SVF 142A29W
- OS : Windows 7
- Prosesor : Intel core I5
- Ethernet : 1/10
- Memori : 4GB
- Tipe : 64-bit

3.2.1.2 Personal Computer Client

Personal computer client bertugas sebagai *client* yang meminta layanan FTP kepada *server*. Komputer *client* akan direpresentasikan dengan nama *host2*. Spesifikasi dari *PC client* adalah:

- Seri : Toshiba Satellite L635
- OS : Windows-8
- Prosesor : Intel Core I5
- Ethernet : 1/10

- Memori : 2.00 GB
- Tipe : 64-bit

3.2.1.3 Router

Router adalah perangkat jaringan yang berada di layer *network*. Berfungsi untuk menyalurkan *traffic data* kepada node-node yang dituju. Router memiliki kemampuan *routing* yang artinya router secara cerdas akan mengetahui kemana tujuan rute perjalanan informasi (*packet*) yang dilewatan. Apakah ditujukan untuk host lain yang satu jaringan ataukah berbeda jaringan. Pada tugas akhir ini sebuah *personal computer* akan dijadikan sebagai *PC router* yang akan dijadikan sebagai *tunnel broker* dan *tunnel server* serta satu buah router sebagai penyambung diantara *client* dan *tunnel server*.. Berikut spesifikasi dari router tersebut:

PC Router :

- Prosesor : Intel Pentium Core I3-4160 3.60 Ghz
- OS : Ubuntu Linux 12.04 LTS
- LAN Card : 10/100 Mbps
- Memori : 2.00 GB
- Tipe : 32-bit

Router:

- Seri : Cisco 1841 revision [7.0]
- DRAM : 262144 Kbytes
- Compact flash : Default 64 MB
Maximum 128 MB
- Slot AIM : One (internal)
- Slot Interface WIC : Two
- Port Konsol : One-up to 115.2 kbps
- Port Auxiliary : One-up to 115.2 kbps

3.2.2 Software (Perangkat Lunak)

Perangkat lunak yang digunakan dalam tugas akhir ini digunakan dalam pengelolaan dan mengakses data yang ada. Serta harus bisa menunjang kebutuhan dalam implementasi sistem. Berikut perangkat lunak yang digunakan.

3.2.2.1 Windows Operating System (OS)

Windows adalah keluarga sistem operasi yang dikembangkan oleh Microsoft dengan menggunakan GUI (*Graphic User Interface*). Sistem Windows telah berevolusi dari MS-DOS sebuah sistem operasi yang berbasis mode teks dan *command-line* hingga sekarang Windows dengan versi terakhir yaitu Windows 8. Pemilihan sistem operasi Windows dalam tugas akhir ini atas dasar pertimbangan bahwa Windows adalah sistem operasi yang sangat populer digunakan sebagai komputer *desktop*, 94,2% pangsa komputer *desktop* dikuasai oleh platform Windows berdasarkan survey yang diadakan oleh marketshare.hitlinks.com.

Selain itu Windows sangat *user friendly* dibandingkan dengan sistem operasi yang lain. Banyaknya *software* yang berbasis Windows dan instalasi *software* yang biasanya lebih mudah juga menjadi keunggulan platform ini. Namun sistem operasi Windows dari segi keamanan masih bisa dibilang kurang.

Untuk perangkat yang menggunakan sistem operasi ini adalaah pada sisi PC server dan PC client. Hal ini disebabkan sebagian besar orang banyak menggunakan sistem operasi ini pada kegiatan sehari-hari.

3.2.2.2 Ubuntu Linux

Ubuntu Linux adalah sistem operasi yang dibuat dengan menggunakan kernel Linux. Ubuntu lahir pada tahun 2004 oleh Canonical. Ubuntu terdiri dari banyak paket yang kebanyakan berasal dari distribusi lisensi *software* bebas.

Untuk tugas akhir ini akan digunakan Ubuntu Linux 12.04 LTS Precise Pangolin. Penggunaan sistem operasi ini berdasarkan pertimbangan dukungan yang dimilikinya terhadap IPv6. Selain itu Ubuntu Linux merupakan distribusi Linux terpopuler dalam penggunaannya sebagai router. Dan yang paling penting adalah sistem operasi ini mendukung dan kompatibel terhadap perangkat lunak jenis lainnya yang digunakan dalam tugas akhir ini.

3.2.2.3 Cisco IOS Image

IOS image adalah file yang berisi seluruh IOS untuk router tersebut. IOS image tergantung pada model router dan fitur dalam IOS. Biasanya fitur yang banyak akan semakin besar pula IOS image nya dan diperlukan kapasitas lebih pada flash dan RAM untuk penyimpanan serta memuat IOS. Pada tugas akhir ini digunakan IOS versi IOS image

12.4-5a yang menunjang kemampuan menjalankan IPv6 dan pembentukan tunnel.

3.2.2.4 Script PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan dalam pembuatan *website* secara dinamis dan dapat dilakukan pembaharuan terhadap *website* tersebut setiap saat. PHP pertama kali dibuat pada tahun 1995 dalam bentuk sekumpulan skrip untuk mengolah data formulir dari web. *Source* kode dari PHP tidak akan ditampilkan di *website* seperti HTML karena PHP diolah dan diproses pada *server*. PHP dapat berjalan pada berbagai macam sistem operasi seperti windows, linux, Mac Os, dan lain-lain.

3.2.2.5 Apache Web Server

Apache adalah *web server* yang dapat dijalankan pada berbagai macam operasi untuk melayani dan memfungsikan situs *web*. Apache akan menampilkan *website* internet seperti menggunakan Mozilla, Opera, atau Chrome berdasarkan kode yang ditulis di dalam *website* tersebut baik menggunakan pemrograman HTML ataupun PHP. Tampilan yang ada didapat berdasarkan *database* pada MySQL. Apache bersifat *open source* sehingga dapat digunakan oleh siapa saja secara gratis.

3.2.2.6 MySQL

MySQL adalah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dan terbuka untuk siapa saja. Setiap orang dapat menggunakannya namun dengan batasan tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL dapat digunakan untuk membuat dan mengelola *database* beserta isinya. Pemanfaatan MySQL adalah dengan menambahkan, mengubah, dan menghapus data pada *database*. Sifat *database* pada MySQL adalah data-data yang dikelola dalam *database* akan diletakkan pada beberapa tabel yang terpisah sehingga pengolahan data jauh lebih cepat.

3.2.2.7 Putty

Putty adalah sebuah program *open source remote console terminal* yang digunakan melakukan *remote* komputer dengan terhubungnya menggunakan port SSH atau sebaliknya. Program ini banyak digunakan oleh para pengguna komputer yang biasanya

digunakan untuk menyambungkan, mensimulasi, atau mencoba berbagai hal yang terkait dengan jaringan. Program ini juga dapat digunakan sebagai *tunnel* di suatu jaringan. Putty digunakan ketika ingin melakukan transfer sebuah data dari sebuah perangkat ke perangkat yang lain dan fungsi bagi penggunaanya dapat menerima data.

3.2.2.8 TFTP Server

TFTP server (*Trivial File transfer protocol*) adalah protokol perpindahan berkas yang sangat sederhana dan protokol ini memiliki fungsionalitas paling dasar dari FTP. Karena sangat sederhana, maka implementasi dari protokol ini dalam komputer yang memiliki memori kecil sangatlah mudah. Dikarenakan kemudahannya tersebut, TFTP pun dimanfaatkan untuk melakukan *booting* komputer seperti halnya router yang tidak memiliki perangkat penyimpanan data. Dalam tugas akhir ini, TFTP server yang digunakan adalah produk dari solarwinds. TFTP tersebut akan dimanfaatkan untuk melakukan *backup* dan penggantian IOS router image router ke flash router.

3.2.2.9 FTP Server

FTP server adalah suatu server yang menjalankan *software* yang berfungsi untuk memberikan layanan tukas menukar file dimana sebuah server harus selalu siap memberikan layanan FTP apabila mendapat permintaan (*request*) dari *FTP client*. FTP server memberikan layanan *download* dan *upload* file antar jaringan TCP/IP yang memanfaatkan *File transfer protocol (FTP)*. FTP merupakan protokol standar untuk melakukan tukar menukar file dalam jaringan yang berjalan pada lapisan aplikasi dari 7 lapisan *OSI Network*.

Pada tugas akhir ini, aplikasi FTP yang digunakan adalah aplikasi Filezilla Server versi 0.9.48. Penggunaan aplikasi Filezilla Server dikarenakan aplikasi ini berjalan pada sistem operasi Windows dan juga mendukung transfer file pada jaringan IPv6.

3.2.2.10 FTP Client

FTP client adalah perangkat yang digunakan untuk melakukan pertukaran data/file antara *FTP client* dan *FTP server*. Pada umumnya *FTP client* digunakan untuk mengunduh ataupun mengupload file ke *FTP server*. Aplikasi ini berjalan berdasarkan protokol FTP (*File Transfer Protocol*) dalam pembentukan sesi koneksi awal sebelum melakukan unggah ataupun unduh file.

Dalam penerapan pada tugas akhir ini, aplikasi yang digunakan adalah keluaran dari Filezilla versi 3.9.0.6. Penggunaan produk ini dikarenakan sangat cepat dan efisien pada penggunaannya. Program ini pun menggunakan sumber daya yang sangat kecil dan menyediakan semua fungsi khas yang diperlukan dalam program FTP. Fitur yang ada juga sangat menunjang, seperti *drag and drop*, antrian transfer, melanjutkan transfer yang terhenti, dan bisa melakukan transfer *file* yang berkapasitas besar. Namun yang terpenting, aplikasi ini menunjang dalam jaringan IPv6.

3.2.2.11 Wireshark

Wireshark adalah salah satu dari sekian banyak *tool network analyzer* yang banyak digunakan oleh *network administrator* dalam menganalisa kinerja jaringan termasuk protokol yang ada di dalamnya. Wireshark disukai disebabkan *interfacenya* yang menggunakan *Graphical User Interface (GUI)* atau tampilan grafis. Wireshark bekerja pada layer terakhir dalam OSI layer, yaitu layer aplikasi. Wireshark dapat membaca data secara langsung dari ethernet, token-ring, FDDI, serial (PPP dan SLIP), 802.11 wireless LAN dan koneksi ATM.

Wireshark mampu menangkap paket-paket data atau informasi yang lewat dalam jaringan. Semua jenis paket informasi dalam berbagai format protokol pun akan lebih mudah untuk ditangkap dan dianalisa. Karenanya tak jarang aplikasi ini dapat dipakai untuk *sniffing* (memperoleh informasi penting seperti *password email* atau *account* lain). Secara umum fungsi dari wireshark adalah memecahkan masalah jaringan, memeriksa keamanan jaringan, men-debug implementasi protokol, mempelajari protokol jaringan internal.

3.3 Instalasi IOS Router

Penggunaan router dalam tugas akhir ini perlu dilakukan pengecekan terlebih sebelum melakukan pengambilan data apakah router yang digunakan sudah mendukung IPv6. Yang pertama kali dilakukan adalah melihat versi dari IOS Cisco pada router. Untuk mengetahuinya dapat memasukkan perintah:

```
Router>show version
```

```

Router>sh version
Cisco IOS Software, 1841 Software (C1841-IPBASEK9-M), Version 12.4(22)T3, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled Tue 01-Sep-09 14:19 by prod_rel_team

ROM: System Bootstrap, Version 12.4(13r)T, RELEASE SOFTWARE (fc1)

Router uptime is 8 minutes
System returned to ROM by power-on
System image file is "flash:c1841-ipbasek9-mz.124-22.T3.bin"

```

Gambar 3.2 Tampilan show version

Dari hasil yang didapat, diketahui IOS yang digunakan pada kedua router adalah “c1841-ipbasek9-mz.124-22.T3.bin”. IOS router tersebut tidak mendukung dalam penggunaan alamat IPv6 dan juga pembuatan *tunnel*. Selain itu juga harus mengetahui berapakah jumlah memori yang ada dan dibutuhkan untuk mengalokasikan IOS yang baru. Untuk mengetahui kapasitas memori dapat diketahui dengan perintah show version.

```

Cisco 1841 (revision 7.0) with 235520K/26624K bytes of memory.
Processor board ID FHK135170JP
2 FastEthernet interfaces
DRAM configuration is 64 bits wide with parity disabled.
191K bytes of NVRAM.
62720K bytes of ATA CompactFlash (Read/Write)

Configuration register is 0x2142

```

Gambar 3.3 Keterangan kapasitas *memory router*

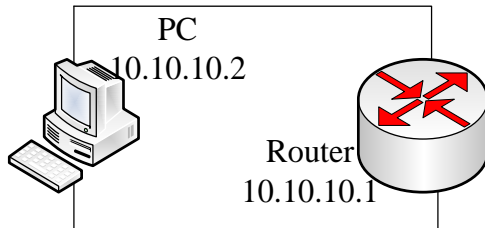
Seperti yang tertera pada perintah show version, kapasitas dari memori DRAM 262145 Kbyte dan untuk Flash yang ada sebesar 62720 Kbytes. Hal ini dapat dijadikan acuan dalam pemilihan IOS yang sesuai dengan penggunaan dan kapasitas dari router tersebut.

IOS router yang diperlukan dapat dicari melalui situs www.cisco.com/cgi-bin/Software/Iosplanner/planner-pool/iosplanner.cgi. IOS router yang dipilih adalah “c1841-adventerprisek9-mz.124-5a.bin” dengan pertimbangan IOS tersebut mampu melakukan konfigurasi tunnel serta mendukung dalam pengalaman IPv6. File tersebut diletakkan pada direktori TFTP server.

Sebelum melakukan perubahan IOS, langkah lebih baiknya melakukan kopi terhadap versi IOS sebelumnya sebagai langkah antisipasi bila IOS tersebut diperlukan dilain waktu. Namun harus dipastikan TFTP server mempunyai koneksi jaringan ke router. Untuk pembentukan koneksi TFTP server-router, akan menggunakan alamat IP dalam satu *network* yang sama dengan memasukkan perintah.

```
Router#configure terminal
Router(config)#interface fa0/0
Router(config-if)#no shutdown
Router(config-if)#ip address 10.10.10.1 255.255.255.0
```

Setelah melakukan *setting* IP pada router, yang harus dilakukan adalah menyesuaikan alamat IP yang ada pada komputer. Alamat IP yang digunakan pada komputer adalah 10.10.10.2 dengan subnet 255.255.255.0 dan *gateway* yang dituju adalah 10.10.10.1. Bila telah terjalin koneksi antar TFTP server dengan router, harus menjalankan aplikasi TFTP server terlebih dahulu.



Gambar 3.4 Konfigurasi *upgrade* IOS

Melakukan backup IOS dan konfigurasi router sebenarnya tidak wajib dilakukan, karena sebenarnya ketika melakukan *upgrade router* tidak berpengaruh terhadap konfigurasi router yang tersimpan dalam *Nonvolatile Random Acces Memory* (NVRAM), akan tetapi yang situasi tidak diinginkan bisa saja terjadi. Perintah yang dilakukan untuk melakukan kopi IOS adalah:

```
Router#copy flash: tftp
Source filename c1841-ipbasek9-mz.124-22.T3.bin
```

```
Address or name of remote host? 10.10.10.2
Destination filename [c1841-ipbasek9-mz.124-22.T3.bin]
```

Sebelum melakukan *upgrade* atau kopi IOS yang baru, perlu dilakukan menghapus IOS sebelumnya. Hal tersebut tidak akan mempengaruhi router, karena IOS ada berada pada RAM ketika awal melakukan *booting*. Perintah menghapus IOS adalah:

```
Router#delete flash: c1841-ipbasek9-mz.124-22.T3.bin
```

Langkah yang dilakukan selanjutnya adalah melakukan kopi IOS yang baru ke dalam router dari TFTP server. Hal ini dilakukan dengan mengetikkan perintah:

```
Router#copy tftp: flash
Source filename ? c1841-adventerprisek9-mz.124-5a.bin
Address or name of remote host ? 10.10.10.2
Destination filename [c1841-adventerprisek9-mz.124-5a.bin]
```

Bila sudah dilakukan kopi IOS, diperlukan tahapan verifikasi file IOS dari TFTP server. Hal ini dimaksudkan untuk mengetahui apakah ada *corrupt* file dalam transfer. Perintah verifikasi dilakukan dengan cara:

```
Router#verify flash: c1841-adventerprisek9-mz.124-5a.bin
```

Tahapan terakhir yang dilakukan adalah mengarahkan *booting system* ke file IOS yang baru dan juga *restart* perangkat router dengan perintah:

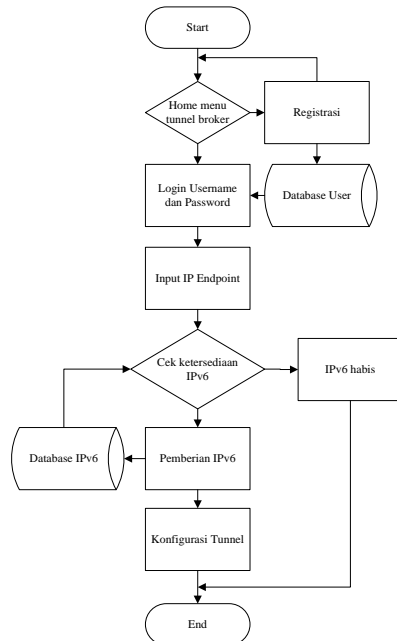
```
Router# config terminal
Router(config)# boot system flash: c1841-adventerprisek9-mz.124-5a.bin
Router(config)#exit
Router#write memory
Router# Reload
```


3.4 Perancangan *Tunnel Broker*

Dalam pembuatan *tunnel broker* dilakukan beberapa langkah, yaitu pembuatan *script* dari *tunnel broker* dan perancangan basis data. *Tunnel broker* akan menggunakan perangkat yang bersifat *open source* sehingga perangkat ini dapat disebarluaskan dan gratis.

3.4.1 Pembuatan *Script PHP*

Skrip PHP dalam tugas akhir ini akan digunakan sebagai sesi antarmuka diantara *client* dengan *tunnel broker* melalui *web*. Skrip PHP akan dirancang agar mampu menjalankan fungsi dari *tunnel broker* yaitu pendaftaran *user*, pengaktifan *tunnel*, penghapusan *tunnel*, dan monitoring jumlah *tunnel* yang masih tersedia. Untuk lebih jelasnya mekanisme desain dari *tunnel broker* dapat dilihat pada gambar 3.5 berikut.



Gambar 3.5 Diagram alir *database tunnel broker*

Gambar 3.5 menjelaskan bahwa ketika *user* melakukan akses melalui *web tunnel broker* *user* akan diberikan pilihan apakah sudah

memiliki akun pendaftaran atau belum, bila belum maka *user* diharuskan melakukan pendaftaran terlebih dahulu. Data kelengkapan *user* akan disimpan pada *database user*. Setelah melakukan registrasi, *user* dapat *login* dan pada halaman awal akan ditunjukkan status dari *user* tersebut terhadap kondisi *tunnel* miliknya, apakah sudah teralokasikan ataukah masih kosong. Langkah yang dilakukan *user* adalah memasukkan *IP endpoint*. *IP endpoint* tersebut akan diperiksa ketersediaan alamat IPv6 pada *database* alamat IPv6. Jika sudah habis maka *user* tidak akan dapat menggunakan alamat IPv6, sedangkan jika masih tersedia maka *user* akan diberikan alamat IPv6 dan juga konfigurasi pengaturan pada sisi *user* agar dapat terkoneksi dengan host IPv6. Untuk skrip yang digunakan pada *tunnel broker* akan diberikan pada bagian lampiran.

3.4.2 Pembuatan Database

Database diperlukan dalam tugas akhir tentang *tunnel broker* sebagai penyimpanan data pengguna. Data dari *user* disimpan agar mudah dilakukan monitoring ataupun verifikasi ketika terjadi masalah di kemudian hari seperti penyalahgunaan *tunnel*. *Software* yang digunakan adalah MySQL. *Software* ini memiliki sistem manajemen basis data yang baik dan terstruktur. Data-data yang ada akan disimpan berdasarkan tabel-tabel yang berbeda dan tidak dijadikan dalam sebuah penyimpanan yang besar namun dipisah-pisah. Walaupun begitu tabel-tabel yang ada dapat dihubungkan berdasarkan keinginan pengguna.

Ada dua buah *database* yang digunakan dalam pembentukan broker, yaitu *database* untuk segala data pengguna dan *database* untuk menyimpan ketersediaan alamat IPv6. Dua *database* tersebut akan saling tersinkronkan berdasarkan fungsinya.

Basis data *user* berfungsi untuk menyimpan segala identitas *user*. Berikut data tabelnya dan tampilannya pada gambar 3.6.

username	password	nama lengkap	alamat	negara	email	ipv4	ipv6
yusro	827ccb0eea8a706c4c34a16891f84e7b	Yusro Muhtadi	Madiun	Indonesia	yusro@muhtadi.com	10.122.1.1	1
admin	21232f297a57a5a743894a0e4a801fc3	Admin Ganteng	Tegal	Indonesia	budi@santoso.com		0
superadmin	1c9c104363b652fb143abd0cb6e2cacc	Gary Almas	Mataram	Indonesia	gary@almas.com	127.0.0.3	2
alex	2043281f3ac876fa0f92af42d53dc704	alexnrudin	jakarta	zambia	ahu@email.com	127.0.0.1	3
yusromuhtadi	d2aefec9dc661bc98eeb6cc12f0b82	Muhammad Yusro Muhtadi	Surabaya	Indonesia	yusromuhtadi@gmail.com	192.168.0.1	4

id: Change Delete Export
 ter rows: Search this table

Gambar 3.6 Database user

username : nama dari pengguna untuk login
 password : kata sandi pengguna untuk login
 namalengkap : nama lengkap dari pengguna
 alamat : alamat pengguna
 negara : negara domisili pengguna
 email : email pengguna
 ipv4 : alamat *endpoint* ipv4 pengguna
 ipv6 : alamat ipv6 yang diberikan ke pengguna

Sedangkan untuk basis data IPv6 berisi tentang alamat IPv6 yang bisa digunakan serta ketersediaannya. Berikut penjelasan data tabel dan tampilannya pada gambar 3.7:

+ Options

	id	ipv6	active
<input type="checkbox"/> Edit Copy Delete	1	2001:B::6/64	1
<input type="checkbox"/> Edit Copy Delete	2	2001:B::5/64	1
<input type="checkbox"/> Edit Copy Delete	3	2001:B::4/64	0
<input type="checkbox"/> Edit Copy Delete	4	2001:B::3/64	1
<input type="checkbox"/> Edit Copy Delete	5	2001:B::1/64	0

☐ Check All With selected: ☐ Change ☐ Delete

Number of rows: 25 Filter rows: Search this table

Gambar 3.7 Basis Data IPv6

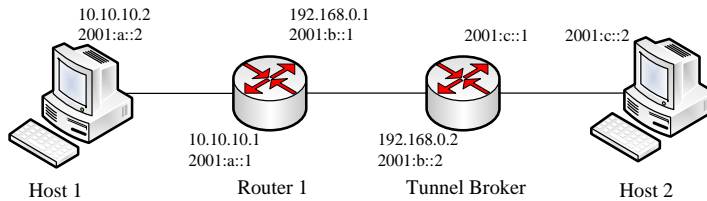
ipv6 : alamat ipv6 yang tersedia
 active : status penggunaan alamat ipv6

3.5 Jaringan IPv6 *Tunnel Broker*

Pada bagian ini akan dijelaskan mengenai perancangan topologi jaringan IPv6 *tunnel broker* serta konfigurasi yang akan digunakan.

3.5.1 Topologi Jaringan

Jaringan Ipv6 *tunnel broker* membutuhkan 4 buah perangkat dalam topologinya. Keempat perangkat tersebut dapat dilihat pada gambar 3.8.



Gambar 3.8 Topologi IPv6 *tunnel broker*

Sedangkan untuk pengalamatan masing-masing perangkat dapat dilihat dalam tabel 3.1 berikut:

Tabel 3.1 Alamat Jaringan IPv6 *tunnel broker*

Perangkat	Interface	Alamat
Host1 (IPv4)	Ethernet	10.10.10.2
Host1 (IPv6)	Ethernet	2001:a::2/64
Router1 (IPv4)	F0/0	10.10.10.1
Router1 (IPv6)	F0/0	2001:a::1/64
Router1 (IPv4)	F0/1	192.168.0.1
Router1 (IPv6)	F0/1	2001:b::1/64
Tunnel Broker (IPv4)	F0/0	192.168.0.2
Tunnel Broker (IPv6)	F0/0	2001:b::2/64
Tunnel Broker (IPv6)	F0/1	2001:c::1/64
Host2	Ethernet	2001:c::2/64

Tunnel broker akan berfungsi sebagai *dual stack router* pada sisi *client* sehingga memiliki dua buah IP pada *interface* f0/0. Pada router1 *interface* 0/1 IP sebenarnya akan didapat ketika sudah dilakukan permintaan aktivasi *tunnel* kepada *tunnel broker*, dan IP yang didapatkan adalah 2001:b::4/64 . Pada sisi host1 dan router1 *interface* 0/0 pada awalnya memiliki IPv4 agar bisa melakukan koneksi ke *web tunnel broker*. Ketika sudah mendapatkan alamat *tunnel*, maka pada host1 dan router1 *interface* 0/0 berganti alamat IPv6 secara manual (IP yang ditentukan oleh *user*). Sedangkan pada sisi router1 *interface* F0/1 diganti dengan mengikuti konfigurasi yang diberikan oleh *tunnel broker*.

3.5.2 Konfigurasi

Pada gambar 3.8 dapat dilihat konfigurasi IPv6 *tunnel broker*. *Server* dan *client* yang memiliki alamat IPv6 akan dipisahkan oleh jaringan IPv4. Berikut keterangan perangkat-perangkat yang digunakan:

1. Host1

Host1 berfungsi sebagai titik awal jaringan dan berlaku sebagai *client* FTP. Host1 akan meminta aktivasi *tunnel* ke *tunnel broker* agar bisa tersambung dengan *FTP server* yang menggunakan IPv6. Host1 juga menjadi *client* dari *tunnel broker*.

2. Router1

Router1 akan berada diantara *FTP client* dan *tunnel broker*. Router1 akan pada awalnya akan menggunakan IPv4. Namun ketika sudah mendapatkan alamat IPv6 dari *tunnel broker*, maka alamatnya akan diganti mengikuti dengan konfigurasi yang diberikan.

3. Tunnel Broker

Tunnel broker akan bekerja ketika adanya permintaan pembuatan, modifikasi, ataupun penghapusan *tunnel*. Host akan meminta pembentukan *tunnel* kepada *tunnel broker* dan *tunnel broker* akan memberikan alamat IPv6 kepada Host1. Transfer *file* akan dilewatkan melalui *tunnel* pada jaringan IPv4.

4. Host2

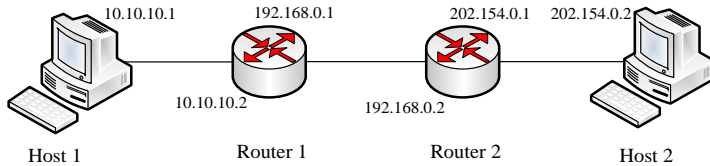
Host2 adalah *server FTP* pada jaringan IPv6. Perangkat ini akan menerima permintaan layanan FTP dari *client* FTP.

3.6 Jaringan IPv4 Murni

Untuk bahan pembandingan analisis performa terhadap jaringan IPv6 *tunnel broker*, maka dirancang sebuah jaringan IPv4 murni. Topologi IPv4 murni akan menggunakan jumlah *node* yang sama dengan IPv6 *tunnel broker*.

3.6.1 Topologi Jaringan

Jaringan IPv4 murni dalam pengimplementasiannya memerlukan 4 buah perangkat yang dapat dilihat pada gambar 3.9.



Gambar 3.9 Topologi IPv4 murni

Untuk pengalaman masing-masing perangkat pada jaringan IPv4 murni, disajikan pada tabel 3.2.

Tabel 3.2 Alamat Jaringan IPv4 murni

Perangkat	Interface	Alamat
Host1	Ethernet	10.10.10.1
Router1	F0/0	10.10.10.2
Router1	F0/1	192.168.0.1
Router2	F0/0	192.168.0.2
Router2	F0/1	202.154.0.1
Host2	Ethernet	202.154.0.2

3.6.2 Konfigurasi

Pada gambar 3.9 dapat dilihat konfigurasi IPv4 murni. Keseluruhan perangkat menggunakan alamat IPv4. Berikut keterangan perangkat-perangkat yang digunakan:

1. Host1

Host1 adalah *node* jaringan IPv4 yang diposisikan akan sebagai FTP *client* dan akan melakukan layanan permintaan FTP kepada FTP *server*.

2. Router1

Router1 akan berlaku sebagai salah satu dari dua buah router pemisah FTP *server* dan FTP *client*. Router1 akan menyalurkan *traffic* diantara *node-node* yang dipisahkannya.

3. Router2

Router2 akan berlaku sebagai salah satu dari dua buah router pemisah *FTP server* dan *FTP client*. Router2 akan menyalurkan *traffic* diantara *node-node* yang dipisahkannya.

4. Host2

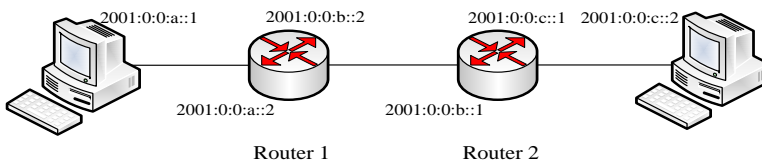
Host2 adalah sebuah host pada jaringan IPv4 yang berfungsi untuk sebagai *server* FTP dan akan melakukan layanan dari permintaan *FTP client*.

3.7 Jaringan IPv6 Murni

Untuk bahan perbandingan yang lain, menggunakan sebuah jaringan yang keseluruhannya menggunakan pengalamatan IPv6. Topologi IPv6 murni akan menggunakan jumlah *node* yang sama dengan jaringan yang lain.

3.7.1 Topologi Jaringan

Jaringan IPv6 murni dalam pengimplementasiannya memerlukan 4 buah perangkat yang dapat dilihat pada gambar 3.10:



Gambar 3.10 Topologi IPv6 murni

Untuk pengalamatan masing-masing perangkat pada jaringan IPv6 murni, disajikan pada tabel 3.3.

Tabel 3.3 Alamat Jaringan IPv6 murni

Perangkat	Interface	Alamat
Host1	Ethernet	2001:0:0:a::1/64
Router1	F0/0	2001:0:0:a::2/64
Router1	F0/1	2001:0:0:b::2/64
Router2	F0/0	2001:0:0:b::1/64
Router2	F0/1	2001:0:0:c::1/64
Host2	Ethernet	2001:0:0:c::2/64

3.7.2 Konfigurasi

Pada gambar 3.10 dapat dilihat konfigurasi IPv6 murni. Keseluruhan perangkat menggunakan alamat IPv6. Berikut keterangan perangkat-perangkat yang digunakan:

1. Host1

Host1 adalah *node* jaringan IPv6 yang diposisikan akan sebagai FTP *client* dan akan melakukan layanan permintaan FTP kepada FTP *server*.

2. Router1

Router1 akan berlaku sebagai salah satu dari dua buah router pemisah FTP *server* dan FTP *client* dengan alamat IPv6. Router1 akan menyalurkan *traffic* diantara *node-node* yang dipisahkannya.

3. Router2

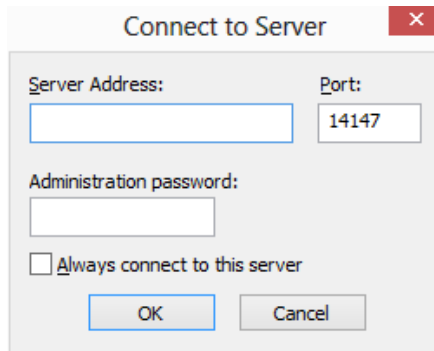
Router2 akan berlaku sebagai salah satu dari dua buah router pemisah FTP *server* dan FTP *client* dengan alamat IPv6. Router2 akan menyalurkan *traffic* diantara *node-node* yang dipisahkannya.

4. Host2

Host2 adalah sebuah host pada jaringan IPv6 yang berfungsi untuk sebagai *server* FTP dan akan melakukan layanan dari permintaan FTP *client*.

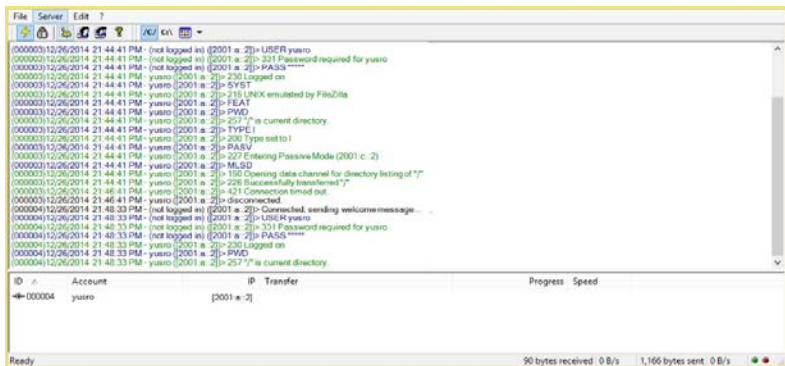
3.8 Pembentukan Server dan Client FTP

Untuk melakukan pengiriman file melalui protokol file transfer, perlu dibangun sebuah *server* sebagai tempat *file* awal berada. *Server* akan melayani permintaan FTP dari *client*, Yang pertama harus dibentuk adalah *server*, karena *server* berfungsi sebagai tempat awal file berada. Selain itu *server* harus dibentuk agar *client* tahu alamat yang akan dituju oleh *client*. Orang-orang yang bisa mengakses *server* sebenarnya dapat siapa saja jika dilakukan pengaturan akses *anonymous*. Namun demi keamanan, alangkah baiknya yang bisa melakukan akses ke *server* dibatasi dengan cara memberikan nama dan *password*.



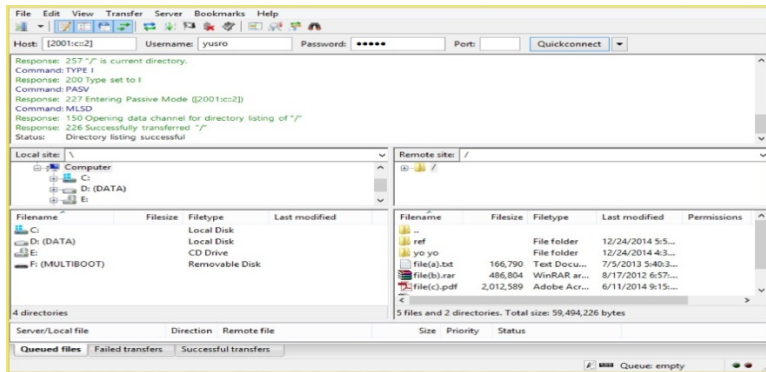
Gambar 3.11 Tampilan awal Filezilla Server

Yang harus dilakukan pertama kali adalah memasukkan alamat *server*. Kemudian, dapat melakukan konfigurasi penambahan *user*, pembatasan folder dan file yang dapat dibagi, pembatasan kecepatan, dan filter terhadap IP. Dalam tugas akhir ini menggunakan *user* yang tidak bersifat *anonymous*.



Gambar 3.12 Filezilla Server ready

Dalam pembentukan FTP client sebenarnya tidak jauh berbeda dengan pembuatan FTP server. Setelah menjalankan aplikasi, kolom *host*, *username*, dan *password* sesuai dengan pengaturan pada FTP server. Bila sudah terhubung maka pada *remote site* akan tampil file dan folder yang dapat diunduh ataupun dimodifikasi.



Gambar 3.13 Filezilla Client

3.9 Metode Pengambilan Data

Pengambilan data dilakukan untuk menguji apakah *IPv6 tunnel broker* dapat bekerja sebagaimana mestinya. Kinerja jaringan dengan menggunakan *IPv6 tunnel broker* juga dapat dilihat dalam pengambilan data ini. Selain itu dapat ditunjukkan pula sebagai perbandingan kinerja dengan antar file berdasarkan ukuran dan tipe dan juga jenis jaringan yang digunakan.

Pengiriman atau pengambilan paket-paket file FTP antara *FTP server* dan *FTP client* akan dianalisis berdasar parameter yang ada. Pada tugas akhir ini akan digunakan lima buah tipe file yang berbeda. Tipe-tipe tersebut adalah:

- file(a).txt dengan ukuran file : 166,790 KB
- file(b).rar dengan ukuran file : 486,804 KB
- file(c).pdf dengan ukuran file : 2.012,589 KB
- file(d).mp3 dengan ukuran file : 5.814,411 KB
- file(e).iso dengan ukuran file : 51.013,632 KB

FTP server akan melakukan transfer file dengan menggunakan dua mode text, yaitu format ASCII dan format *binary*. Secara *default*, sebenarnya FTP menggunakan mode ASCII dalam transfer data. Untuk merepresentasikan mode ASCII maka digunakan file(a).txt dengan ukuran file sebesar 166,790 KB. Dalam merepresentasikan mode transfer *binary* digunakan empat buah tipe file yang lain, yaitu rar, pdf, mp3, iso. Pemilihan tipe rar dikarenakan banyak *FTP server* akan

melakukan kompresi ukuran filenya untuk menghemat dari kapasitas penyimpanan atau bisa disebut mode binary *compressed*. Selain itu format rar adalah salah satu format kompresi yang paling populer digunakan. Tipe file pdf dipilih karena merupakan tipe file yang banyak digunakan dalam lingkungan kampus dan menggunakan mode binary *uncompressed*. Format tipe mp3 dipilih karena tipe tersebut merupakan file tipe yang paling banyak digunakan, disimpan, dan diunduh pada FTP server[7] dan menggunakan mode binary *uncompressed*. Sedangkan format file terakhir yaitu iso digunakan karena format tersebut dapat merepresentasikan file dengan ukuran besar dan tipe ini menggunakan mode binary *uncompressed*. Dari kelima buah file tersebut tidak berada dalam konteks perbedaan file, namun lebih kearah perbedaan ukuran. Hal ini dikarenakan untuk melihat pengaruh perbedaan ukuran file pada saat melakukan transfer data pada aplikasi *file transfer protocol*. Sedangkan untuk perbedaan tipe file, akan digunakan dua buah file dengan ukuran yang hampir sama namun tipe file yang berbeda yaitu:

- file(x).txt dengan ukuran file : 132,991 KB
- file(y).pdf dengan ukuran file : 132,876 KB

Untuk tipe txt akan merepresentasikan file dengan format ASCII dan tipe file pdf akan merepresentasikan file dengan format binary.

Masing-masing file akan dilakukan proses transfer dari FTP server ke FTP client sebanyak 10 kali setiap topologi yang digunakan. Jadi secara keseluruhan akan ada 210 kali pengambilan data yang dilakukan.

BAB 4

ANALISIS DATA

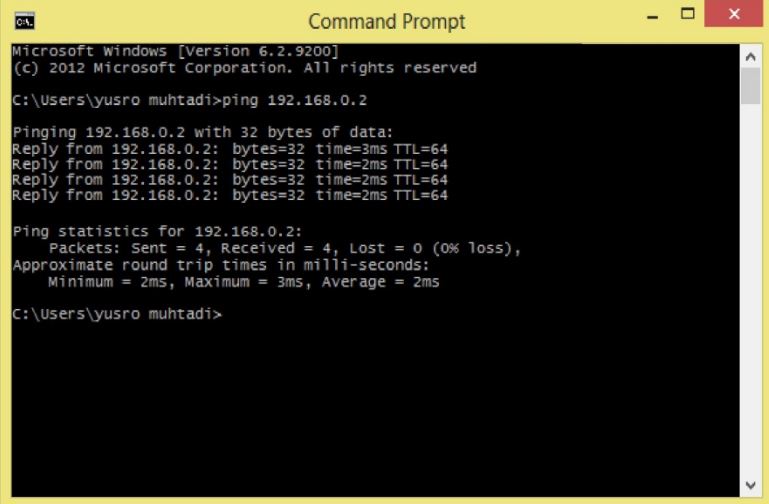
Pada bab ini akan dibahas tentang hasil dari koneksi jaringan, pengambilan data, pembentukan *tunnel broker* analisis performa pada parameter *latency* dan *throughput*.

4.1 Analisis Jaringan

Dalam konfigurasi jaringan, perlu dilakukan pemeriksaan terhadap *node* jaringan dari *client* sampai dengan *server*. Untuk mendapatkan gambaran tentang interkoneksi jaringan dapat dilakukan dengan melakukan ping dan traceroute. Topologi yang digunakan dalam pengujian adalah topologi bus.

4.1.1 Analisis Jaringan IPv6 Tunnel Broker

Pada jaringan IPv6 *tunnel broker*, untuk inisialisasi awalnya adalah dengan menggunakan pengalamatan IPv4 dari *host1*, *router1*, dan *tunnel broker*. Sebelum melakukan permintaan aktivasi *tunnel*, perlu dilakukan uji koneksi terlebih dahulu dari *host1* ke *tunnel broker*.



```
Microsoft Windows [version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved

C:\Users\yusro muhtadi>ping 192.168.0.2

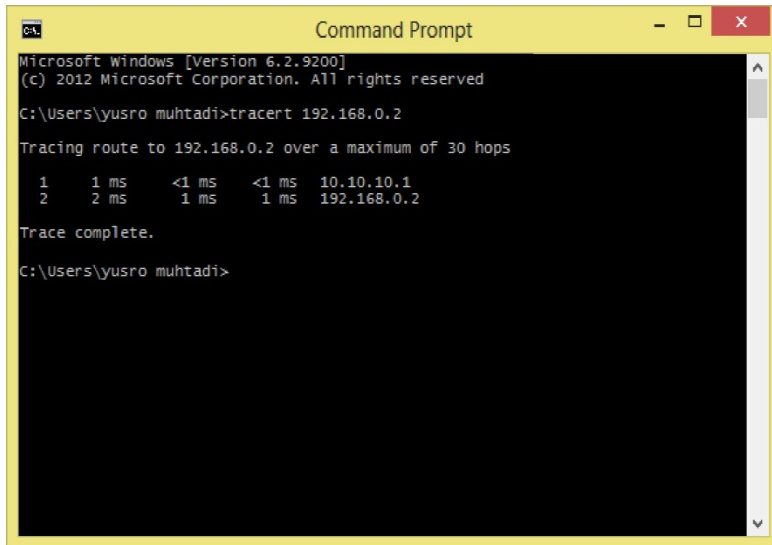
Pinging 192.168.0.2 with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time=3ms TTL=64
Reply from 192.168.0.2: bytes=32 time=2ms TTL=64
Reply from 192.168.0.2: bytes=32 time=2ms TTL=64
Reply from 192.168.0.2: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 3ms, Average = 2ms

C:\Users\yusro muhtadi>
```

Gambar 4.1 Ping host1 ke *tunnel broker*

Pada gambar 4.1 dapat dilihat bahwa host1 mendapatkan balasan dari *tunnel broker*. Artinya *node* dari host1 sudah terkoneksi dengan *tunnel broker*. Yang perlu dilakukan selanjutnya adalah melihat hop yang dilewati dari host1 ke *tunnel broker* dengan cara melakukan *tracert*.



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved

C:\Users\yusro muhtadi>tracert 192.168.0.2

Tracing route to 192.168.0.2 over a maximum of 30 hops

  1    1 ms    <1 ms    <1 ms    10.10.10.1
  2    2 ms    1 ms     1 ms    192.168.0.2

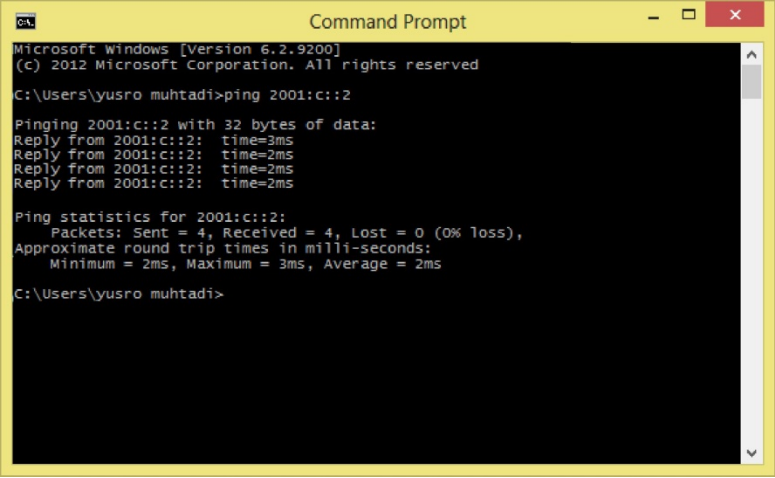
Trace complete.

C:\Users\yusro muhtadi>
```

Gambar 4.2 Traceroute host1 ke *tunnel broker*

Dari gambar 4.2 dapat dilihat bahwa paket akan dilewatkan melalui 2 hop dengan alamat IPv4. Paket akan melalui *gateway* yang berada pada router1 (10.10.10.1). Kemudian paket diteruskan ke *tunnel broker* yang memiliki alamat (192.168.0.2).

Setelah melakukan aktivasi *tunnel* dan melakukan konfigurasi, langkah yang sama perlu dilakukan. Yaitu menguji interkoneksi jaringan dari host1 sebagai *client* dengan host2 sebagai *server* FTP dengan melakukan ping.



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved

C:\Users\yusro muhtadi>ping 2001:c::2

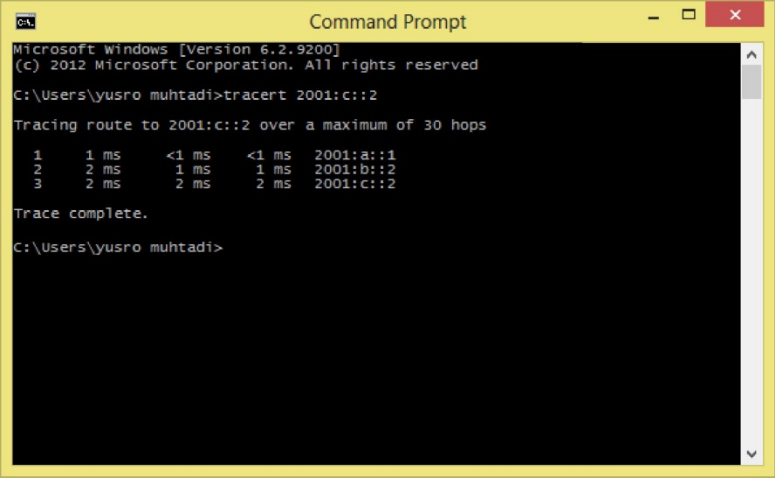
Pinging 2001:c::2 with 32 bytes of data:
Reply from 2001:c::2: time=3ms
Reply from 2001:c::2: time=2ms
Reply from 2001:c::2: time=2ms
Reply from 2001:c::2: time=2ms

Ping statistics for 2001:c::2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 3ms, Average = 2ms

C:\Users\yusro muhtadi>
```

Gambar 4.3 Ping jaringan IPv6 *tunnel broker*

Terlihat di gambar 4.3 bahwa dari host1 mendapatkan balasan dari host2. Ini membuktikan bahwa jaringan tersebut sudah terkoneksi. Kemudian dilakukan pengecekan jalur paket yang akan dikirimkan dengan melakukan traceroute dari host1 ke host2.



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved

C:\Users\yusro muhtadi>tracert 2001:c::2

Tracing route to 2001:c::2 over a maximum of 30 hops
  0  0 ms  <1 ms  <1 ms  2001:a::1
  1  1 ms   1 ms   1 ms  2001:b::2
  2  2 ms   2 ms   2 ms  2001:c::2

Trace complete.

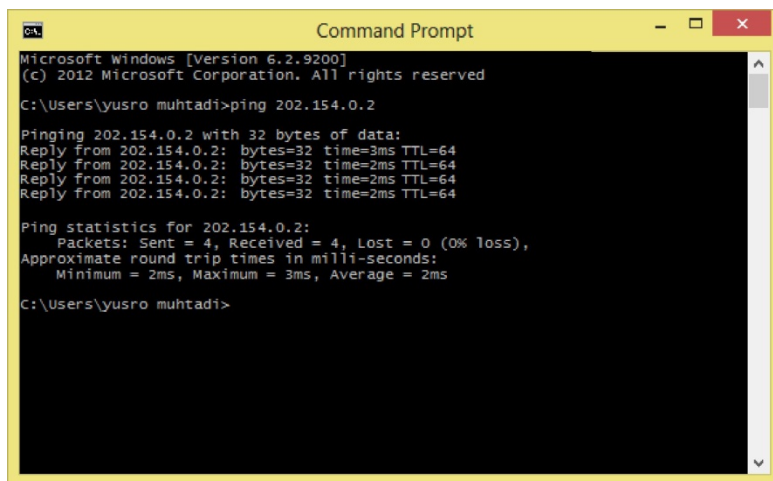
C:\Users\yusro muhtadi>
```

Gambar 4.4 Traceroute jaringan IPv6 *tunnel broker*

Pada gambar 4.4 terlihat bahwa semua alamat yang ada sudah berganti menjadi IPv6. Rute paket untuk mencapai host adalah dengan melewati *gateway router1* (2001:a::1), diteruskan kepada *tunnel broker* dengan alamat (2001:b::2), dan akhirnya paket sampai kepada *host2* yang memiliki alamat 2001:c::2. Host1, router1, dan *tunnel broker* berganti alamat dengan IPv6 (berbeda dengan traceroute host1 ke *tunnel broker*) dikarenakan sudah ada pembentukan *tunnel* dan konfigurasi perubahan alamat IPv6 sebelumnya.

4.1.2 Analisis Jaringan IPv4

Dalam konfigurasi jaringan IPv4, semua perangkat akan disusun sesuai dengan jumlah hop yang sama seperti pada pengujian jaringan IPv6 *tunnel broker*. Semua perangkat baik pada sisi host ataupun router akan diberikan alamat IPv4. Jenis *routing* yang digunakan adalah *static routing*, maka proses pada router yang terjadi hanyalah *routing* dan *forwarding* seperti jaringan pada umumnya. Perlu dilakukan ping dan traceroute untuk mengetahui konektivitas jaringan ini.



```
Command Prompt
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved
C:\Users\yusro muhtadi>ping 202.154.0.2

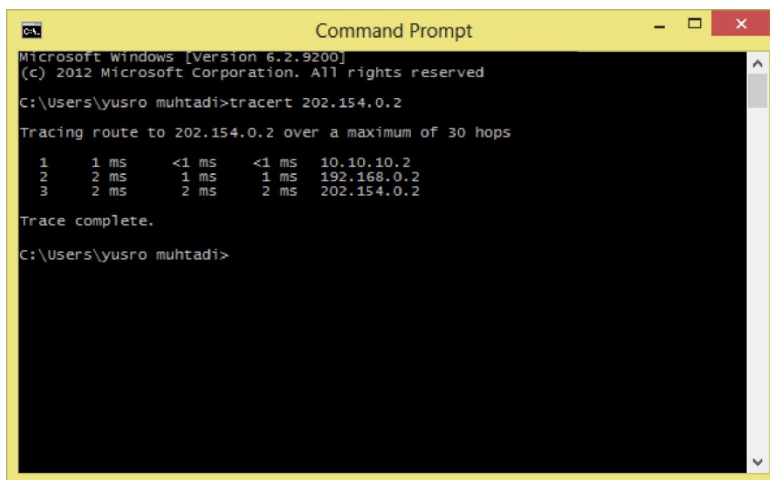
Pinging 202.154.0.2 with 32 bytes of data:
Reply from 202.154.0.2: bytes=32 time=3ms TTL=64
Reply from 202.154.0.2: bytes=32 time=2ms TTL=64
Reply from 202.154.0.2: bytes=32 time=2ms TTL=64
Reply from 202.154.0.2: bytes=32 time=2ms TTL=64

Ping statistics for 202.154.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 3ms, Average = 2ms
C:\Users\yusro muhtadi>
```

Gambar 4.5 Ping jaringan IPv4

Dari gambar 4.5 terlihat bahwa host1 sudah mendapatkan balasan dari host2. Ini menunjukkan bahwa host1 dan host2 sudah terhubung.

Sedangkan untuk melihat jalur paket yang dilewati perlu dilakukan traceroute jaringan tersebut.



```
Microsoft Windows [version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved

C:\Users\yusro muhtadi>tracert 202.154.0.2

Tracing route to 202.154.0.2 over a maximum of 30 hops

  1    1 ms    <1 ms    <1 ms    10.10.10.2
  2    2 ms    1 ms     1 ms    192.168.0.2
  3    2 ms    2 ms     2 ms    202.154.0.2

Trace complete.

C:\Users\yusro muhtadi>
```

Gambar 4.6 Traceroute jaringan IPv4

Pada gambar 4.6 terlihat bahwa paket akan dilewatkan pada *gateway* router1 (10.10.10.2), kemudian *gateway* router2 (192.168.0.2), dan paket sampai pada host2 dengan alamat 202.154.0.2. Semua alamat yang ada menggunakan alamat IPv4 dan sudah terhubung antara host1 dan host2.

4.1.3 Analisis Jaringan IPv6

Konfigurasi jaringan IPv6 sebenarnya tidak jauh berbeda dengan konfigurasi jaringan sebelumnya dengan jumlah hop yang sama. Namun yang membedakan adalah pengalamatannya yang menggunakan IPv6. Jenis *routing* juga sama, hanya ada *routing* dan *forwarding* seperti paket pada umumnya. Untuk menguji jaringan ini dilakukan mekanisme ping dan traceroute.


```
Command Prompt
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved

C:\Users\yusro muhtadi>ping 2001:0:0:c::2

Pinging 2001:0:0:c::2 with 32 bytes of data:
Reply from 2001:0:0:c::2:  time=3ms
Reply from 2001:0:0:c::2:  time=2ms
Reply from 2001:0:0:c::2:  time=2ms
Reply from 2001:0:0:c::2:  time=2ms

Ping statistics for 2001:0:0:c::2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 3ms, Average = 2ms

C:\Users\yusro muhtadi>
```

Gambar 4.7 Ping jaringan IPv6

Dari gambar 4.7 terlihat bahwa host1 sudah mendapatkan balasan dari host2. Ini menunjukkan bahwa host1 dan host2 sudah terhubung. Sedangkan untuk melihat jalur paket yang dilewati perlu dilakukan traceroute jaringan tersebut.

```
Command Prompt
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved

C:\Users\yusro muhtadi>tracert 2001:0:0:c::2

Tracing route to 2001:0:0:c::2 over a maximum of 30 hops

  1    1 ms    <1 ms    <1 ms    2001:0:0:a::2
  2    2 ms     1 ms     1 ms    2001:0:0:b::1
  3    2 ms     2 ms     2 ms    2001:0:0:c::2

Trace complete.

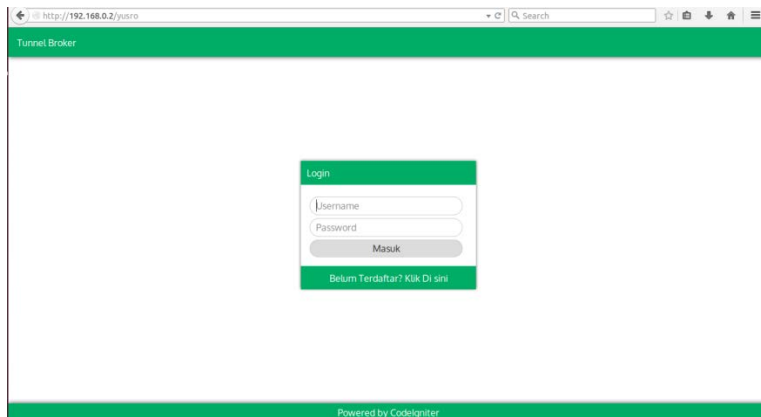
C:\Users\yusro muhtadi>
```

Gambar 4.8 Traceroute jaringan IPv6

Pada gambar 4.8 terlihat bahwa paket akan dilewatkan pada *gateway* router1 (2001:0:0:a::2), kemudian *gateway* router2 (2001:0:0:b::1), dan paket sampai pada host2 dengan alamat 2001:0:0:c::2. Semua alamat yang ada menggunakan alamat IPv6 dan sudah terhubung antara host1 dan host2.

4.2 Tampilan *Web Tunnel Broker*

Untuk melakukan hubungan dari host1 yang memiliki alamat IPv4 dengan host2 yang beralamat IPv6, perlu dilakukan konfigurasi mekanisme transisi IPv4 ke IPv6. *Client* akan mendapatkan konfigurasi dan alamat tersebut dengan cara mengakses kepada *web tunnel broker*. Alamat *tunnel broker* pada tugas akhir ini menggunakan alamat <http://192.168.0.2/yusro>.



Gambar 4.9 Tampilan *home tunnel broker*

Pada gambar 4.9 tampak tampilan awal dari *tunnel broker*. *User* akan diberikan dua buah pilihan, yaitu jika sudah terdaftar dan memiliki akun *tunnel broker* bisa langsung memasukkan *username* dan *password* agar bisa mendapatkan alamat dan konfigurasi IPv6. Sedangkan jika ada pengguna yang belum memiliki *username*, maka diharuskan untuk mendaftar terlebih dahulu dengan mengisi identitas pribadi.

Registration form fields:

- Username
- Password
- Ulangi Password
- Nama Lengkap
- Alamat
- Negara
- Email
- Submit

Gambar 4.10 Tampilan daftar baru

Gambar 4.10 menampilkan halaman pendaftaran pengguna baru. Pengguna harus mengisi identitas berupa *username*, *password*, konfirmasi *password*, nama lengkap, alamat, negara, dan alamat email. Semua identitas tersebut akan dimasukkan ke dalam *database user*.

User Status Information:

My Status	
Nama Lengkap	Muhammad Yusro Muhtadi
Alamat	Surabaya
Negara	Indonesia
Email	yusromuhtadi@gmail.com
IPv4	192.168.0.1
IPv6	Tidak Ada Data

Gambar 4.11 Tampilan status pengguna

Setelah masuk kedalam *tunnel broker*, *user* akan diberikan tampilan status kondisi jaringannya sekarang seperti pada gambar 4.11 yaitu terkait nama, alamat, negara email, alamat IPv4 *endpoint*-nya dan

juga alamat IPv6 yang diberikan oleh *tunnel broker*. Yang perlu dilakukan agar kita bisa mendapatkan alamat IPv6 dan konfigurasinya adalah memasukkan alamat IPv4 *endpoint* kita. Kemudian *tunnel broker* akan memberikan alamat IPv6 beserta konfigurasinya seperti pada gambar 4.12. Pada tugas akhir ini konfigurasi yang diberikan hanya terbatas pada konfigurasi router cisco.

http://192.168.0.2/yusra/index.php/home/config

Tunnel Broker

Config

Ketikkan IP anda untuk memperoleh konfigurasi IPv6.

IP Anda

Submit

IPv4	192.168.0.1
IPv6 Del	2001:B::1/64

Ketik kode berikut pada CMD anda untuk konfigurasi:

```
configure terminal
interface tunnel0
no ip address
ipv6 enable
ipv6 address 2001:B::1/64
tunnel source 192.168.0.1
tunnel destination 192.168.0.2
tunnel mode ipv6ip
ipv6route ::0 tunnel0
end
```

Powered by CodeIgniter

Gambar 4.12 Tampilan konfigurasi

4.3 Analisis Jaringan IPv6 *Tunnel Broker*

Pada jaringan IPv6 *tunnel broker* akan dilakukan analisis terhadap *latency* dan *throughput* yang didapat. *Latency* dan *throughput* tersebut akan dianalisis dalam setiap pengirimannya.

4.3.1 Analisis *Latency* Jaringan IPv6 *Tunnel Broker*

Analisis pengolahan data yang pertama adalah *latency* yang dinyatakan dalam satuan sekon atau detik. *Latency* adalah waktu yang dibutuhkan untuk menyelesaikan suatu koneksi. *Latency* akan dimulai dari waktu pertama kali *client* FTP mengirimkan permintaan sampai hasil file tersebut diterima oleh *client* FTP. *Latency* yang terjadi pada proses ini akan meliputi dari *latency* pada *client* (waktu yang dibutuhkan *client* dalam meminta file FTP), *latency* pada *server* FTP

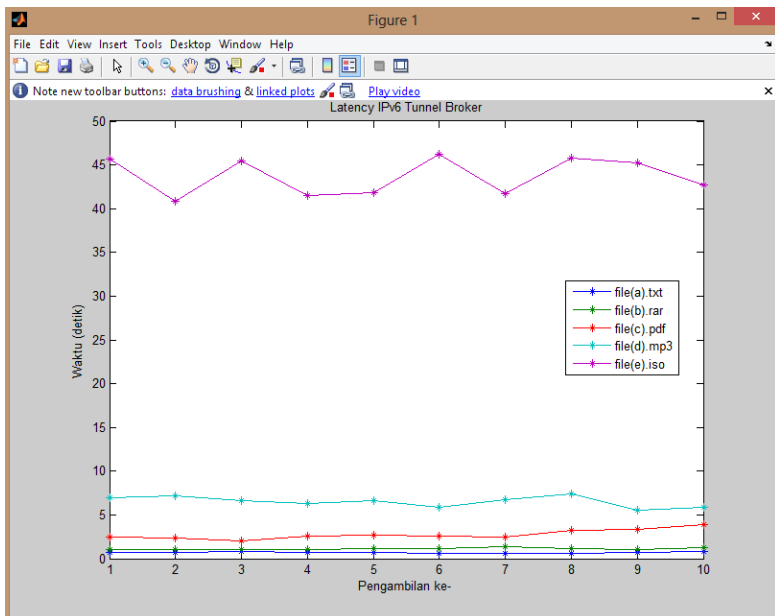
(waktu yang dibutuhkan untuk sebuah server FTP untuk melakukan proses permintaan FTP), dan *latency* pada jaringan (waktu yang dibutuhkan oleh paket-paket data dari *client* menuju ke *server* atau bisa juga sebaliknya). *Latency* menjadi salah satu poin penekanan dalam melakukan transfer data.

Hasil dari nilai *latency* pengiriman lima file pada jaringan IPv6 *tunnel broker* dapat dilihat pada tabel 4.1 berikut:

Tabel 4.1 *Latency IPv6 tunnel broker* (detik)

Pengam bilan ke	File (a).txt	File (b).rar	File (c).pdf	File (d).mp3	File (e).iso
1	0.671	1.05	2.501	7.008	45.616
2	0.714	1.075	2.337	7.165	40.871
3	0.777	1.038	2.043	6.672	45.463
4	0.694	0.994	2.562	6.251	41.534
5	0.704	1.125	2.682	6.584	41.873
6	0.635	1.106	2.555	5.914	46.244
7	0.623	1.375	2.421	6.695	41.702
8	0.652	1.186	3.272	7.368	45.805
9	0.725	1.006	3.331	5.584	45.257
10	0.782	1.287	3.907	5.826	42.73
Rata-rata	0.6977	1.1242	2.7611	6.5067	43.7095

Untuk memudahkan analisis, data akan dijadikan kedalam bentuk grafik seperti pada gambar 4.13.



Gambar 4.13 Latency IPv6 tunnel broker

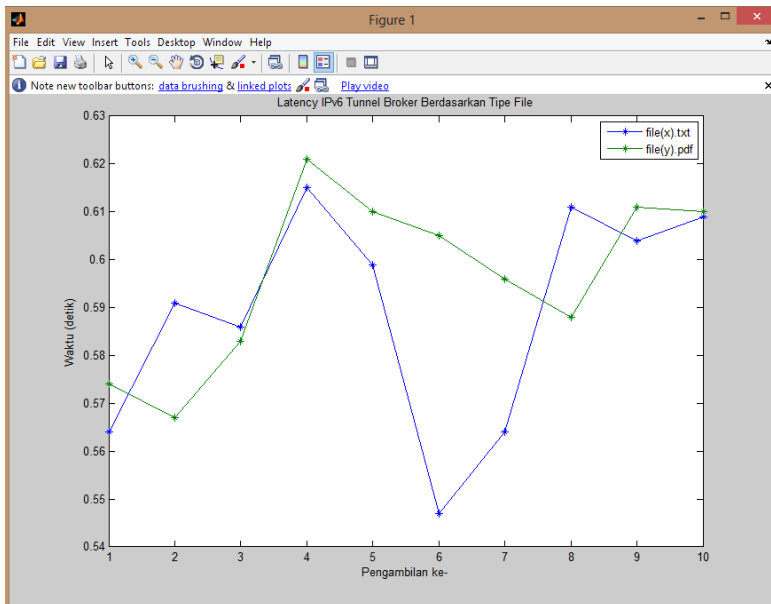
Pada tabel 4.1 dan gambar 4.13 untuk file(a).txt yang berukuran sebesar 166.79 Kb rentang nilai *latency* yang ada adalah berada pada 0,623 – 0,782 detik. File(b).rar dengan *latency* terkecil sebesar 0,994 detik dan terbesar dengan nilai 1,375 detik. Pada file yang ketiga yaitu file(c). pdf memiliki nilai *latency* dari 2,403 detik sampai 3,907 detik. File keempat yaitu file dengan tipe mp3 yang berukuran 5.814 KB memiliki *latency* terkecil sebesar 5,584 detik dan *latency* terbesarnya adalah 7,368 detik. Sedangkan file yang memiliki ukuran paling besar file(e).iso dengan nilai *latency* 40,871 detik dan terbesar senilai 46,244 detik.

Semakin besar ukuran file yang dikirimkan juga akan berpengaruh terhadap nilai parameter *latency*. Selain itu rentang terbesar juga dimiliki oleh ukuran file yang paling besar yaitu file(e).iso. *Latency* juga berbanding lurus dengan ukuran file yang ada. Semakin besar ukuran file maka *latency* juga akan semakin besar, begitupula sebaliknya.

Untuk pengambilan data *latency* pada perbandingan tipe file IPv6 tunnel broker dapat dilihat pada tabel 4.2 dan gambar 4.14.

Tabel 4.2 Latency IPv6 tunnel broker berdasarkan tipe file (detik)

Pengambilan Ke	File(x).txt	File(y).pdf
1	0.564	0.574
2	0.591	0.567
3	0.586	0.583
4	0.615	0.621
5	0.599	0.61
6	0.547	0.605
7	0.564	0.596
8	0.611	0.588
9	0.604	0.611
10	0.609	0.61
Rata-rata	0.589	0.5965



Gambar 4.14 Latency IPv6 tunnel broker berdasarkan tipe file

Dapat dilihat dalam pengambilan data pada file(x).txt memiliki *latency* terkecil 0,547 detik dan *latency* terbesar pada nilai 0,615 detik. Sedangkan untuk file(y).pdf memiliki rentang *latency* dari 0,567 detik sampai dengan 0,621 detik. Tidak ada perbedaan *latency* yang terlalu jauh dari kedua file tersebut, karena ukuran file perbedaannya sangat kecil dan dapat diketahui perbedaan tipe file tidak mempengaruhi performa pada aplikasi FTP dalam jaringan IPv6 *tunnel broker*.

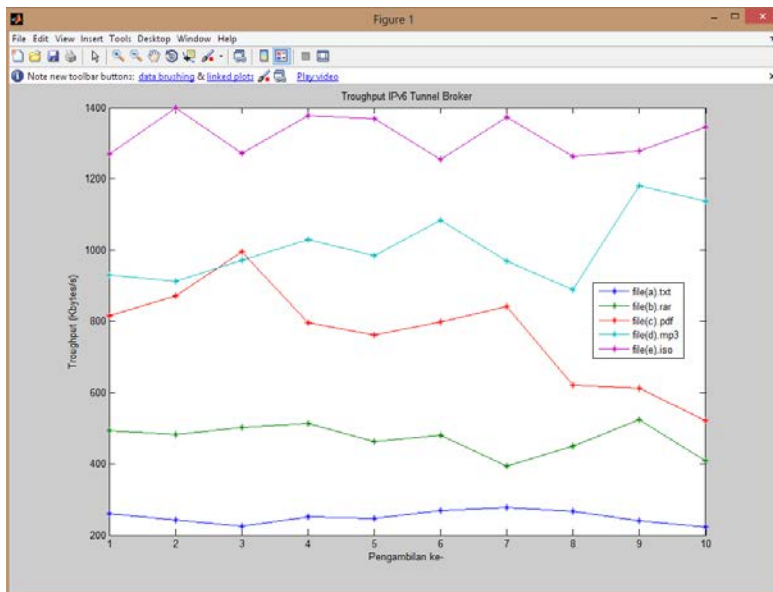
4.3.2 Analisis *Troughput* Jaringan IPv6 *Tunnel Broker*

Analisis pengolahan data yang kedua adalah mengenai parameter *troughput*. *Troughput* merupakan kecepatan transfer rata-rata dari suksesnya sebuah paket yang dikirimkan setiap detiknya. Pengambilan dilakukan dengan cara *download* file dari *server* FTP ke *client* FTP. Disaat yang bersamaan, *client* akan melakukan *captured* data yang masuk. Berikut hasil pengambilan data yang disajikan dalam tabel 4.3.

Tabel 4.3 *Troughput* IPv6 *tunnel broker* (Kbytes/s)

Pengambilan ke	File (a).txt	File (b).rar	File (c).pdf	File (d).mp3	File (e) .iso
1	258.5693	493.622857	814.713715	929.625893	1268.3276
2	242.59944	482.85093	871.184852	911.562884	1398.16207
3	224.658945	501.982659	995.114537	971.454478	1272.0912
4	250.331412	511.742455	795.553864	1030.11693	1378.24788
5	246.917614	462.744667	760.40604	983.152242	1368.25418
6	267.661417	480.178282	797.706067	1083.16247	1253.17056
7	277.720706	394.089273	824.15763	968.452497	1371.91912
8	265.812883	450.478685	621.094438	889.16273	1263.72318
9	240.055172	523.970596	612.19964	1181.26246	1277.19871
10	223.286445	408.247086	521.12388	1138.01081	1343.85986
Rata-rata	249.761333	470.990749	763.125467	1008.59634	1319.49544

Untuk memudahkan analisis, data akan dijadikan kedalam bentuk grafik seperti pada gambar 4.15.



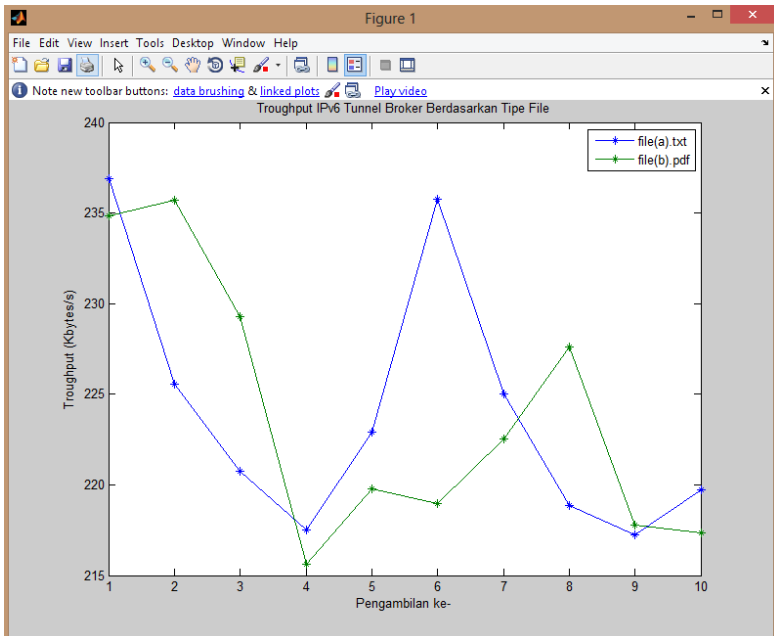
Gambar 4.15 Troughput IPv6 tunnel broker

Dari tabel 4.3 dan gambar 4.15 didapatkan hasil File(a).txt nilai minimum *troughput*-nya adalah 223,2864 KB/s dan nilai maksimumnya 277,7207 Kbytes/s. Pada file dengan format mode *binary* yaitu file(b).rar memiliki nilai *troughput* dengan rentang 394,0893 KB/s sampai dengan 523,9706 Kbytes/s. File ketiga yaitu file(c).pdf memiliki nilai *troughput* terkecil sebesar 521,1239 Kbytes/s dan nilai terbesar 995,1145 Kbytes/s. File(d).mp3 mempunyai nilai *troughput* 889,1627 KB/s sampai dengan 1.181,262 Kbytes/s. Dan file mode *binary* yang memiliki ukuran paling besar yaitu file(e).iso memiliki nilai *troughput* minimal 1.253,171 Kbytes/s sampai dengan 1.398,162 Kbytes/s.

Untuk pengambilan data *troughput* pada perbandingan tipe file IPv6 tunnel broker dapat dilihat pada tabel 4.4 dan gambar 4.16.

Tabel 4.4 *Throughput IPv6 tunnel broker berdasarkan tipe file (Kbytes/s)*

Pengambilan Ke	File(x).txt	File(y).pdf
1	236.9003546	234.8397213
2	225.5780034	235.70194
3	220.7339934	229.2607204
4	217.5373984	215.6423076
5	222.8866444	219.7537377
6	235.7345338	218.9603306
7	225.0008013	222.5123879
8	218.8548118	227.6143728
9	217.2469479	217.8003273
10	219.7249639	217.357377
Rata-rata	224.0198453	223.9443223



Gambar 4.16 *Troughput IPv6 tunnel broker berdasarkan tipe file*

Dapat dilihat dalam pengambilan data pada file(x).txt memiliki *throughput* terkecil 217,24 Kbytes/s dan *throughput* terbesar pada nilai 236,9 Kbytes/s. Sedangkan untuk file(y).pdf memiliki rentang *throughput* dari 215,642 Kbytes/s sampai dengan 235,70194 Kbytes/s. Tidak ada perbedaan *throughput* yang terlalu jauh dari kedua file tersebut, karena ukuran file perbedaannya sangat kecil dan dapat diketahui perbedaan tipe file tidak mempengaruhi performa pada aplikasi FTP dalam jaringan IPv6 *tunnel broker*.

4.4 Analisis Jaringan IPv4

Sama dengan analisis jaringan IPv6 *tunnel broker*, pada jaringan ini juga akan dilakukan analisa terhadap *latency* dan *throughput* pengiriman file melalui FTP.

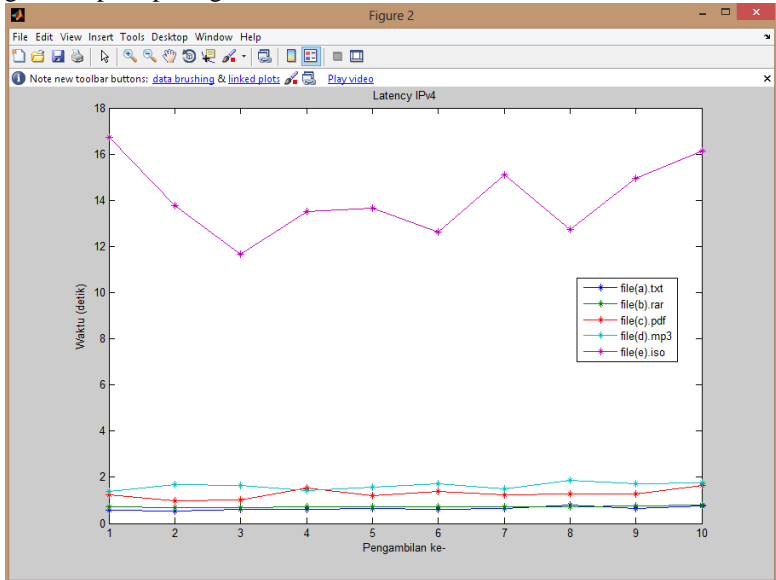
4.4.1 Analisis *Latency* Jaringan IPv4

Analisis pengolahan data yang pertama adalah *latency* yang dinyatakan dalam satuan sekon atau detik. Hasil dari nilai *latency* lima file dapat dilihat pada tabel 4.5.

Tabel 4.5 *Latency* jaringan IPv4 (detik)

Pengam bilan ke	File (a).txt	File (b).rar	File (c).pdf	File (d).mp3	File (e).iso
1	0.56	0.71	1.23	1.37	16.741
2	0.527	0.699	0.993	1.7	13.753
3	0.6	0.684	1.02	1.631	11.671
4	0.611	0.726	1.551	1.41	13.513
5	0.631	0.715	1.187	1.58	13.662
6	0.597	0.733	1.374	1.722	12.616
7	0.636	0.703	1.221	1.515	15.111
8	0.792	0.718	1.273	1.88	12.728
9	0.652	0.741	1.284	1.724	14.941
10	0.75	0.807	1.631	1.741	16.145
Rata- rata	0.6356	0.7236	1.2764	1.6273	14.0881

Untuk memudahkan analisis, data akan dijadikan kedalam bentuk grafik seperti pada gambar 4.17.



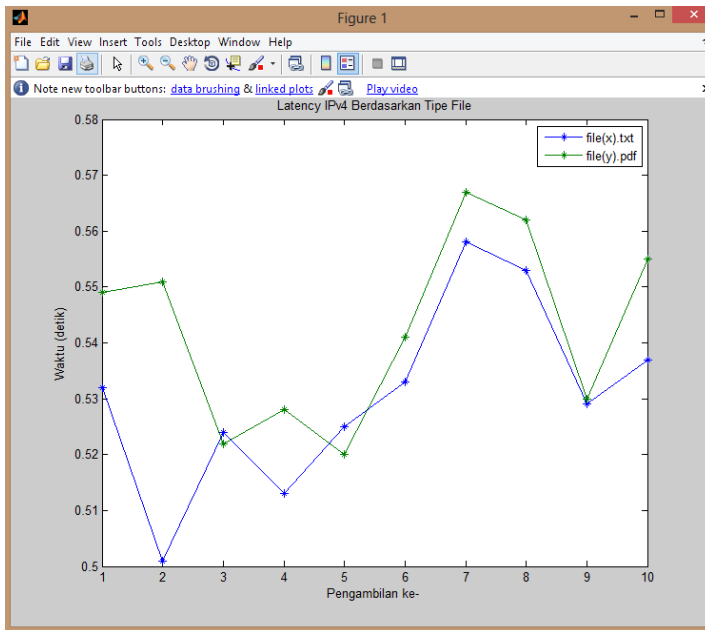
Gambar 4.17 Latency jaringan IPv4

Pada tabel 4.4 dan gambar 4.17 dapat dilihat bahwa nilai terkecil file(a).txt sebesar 0,527 detik dan terbesar senilai 0,792 detik. Pada file(b).rar nilai *latency* berada pada rentang 0,684 detik sampai dengan 0,807 detik. File(c).pdf memiliki nilai *latency* terkecil sebesar 0,993 detik dan terbesar 1,63 detik. File keempat yaitu file(d).mp3 mempunyai nilai *latency* dari 1,37 detik untuk nilai terkecil dan 1,88 detik untuk nilai terbesar. Dan untuk file yang terakhir memiliki nilai *latency* yang berbeda jauh bila dibandingkan dengan file-file yang lain yaitu dari 11,671 detik sampai 16,741 detik. Hal ini dikarenakan ukuran file yang sangat besar. Nilai *latency* yang didapat akan linear dengan ukuran paket yang dikirim. Semakin besar ukuran paket, maka *latency* juga akan semakin besar.

Untuk pengambilan data *latency* pada perbandingan tipe file IPv4 murni dapat dilihat pada tabel 4.6 dan gambar 4.18.

Tabel 4.6 Latency IPv4 murni berdasarkan tipe file (detik)

Pengambilan Ke	File(x).txt	File(y).pdf
1	0.532	0.549
2	0.501	0.551
3	0.524	0.522
4	0.513	0.528
5	0.525	0.52
6	0.533	0.541
7	0.558	0.567
8	0.553	0.562
9	0.529	0.53
10	0.537	0.555
Rata-rata	0.5305	0.5425



Gambar 4.18 Latency IPv4 berdasarkan tipe file

Dapat dilihat dalam pengambilan data pada file(x).txt memiliki *latency* terkecil 0,501 detik dan *latency* terbesar pada nilai 0,558 detik. Sedangkan untuk file(y).pdf memiliki rentang *latency* dari 0,52 detik sampai dengan 0,567 detik. Tidak ada perbedaan *latency* yang terlalu jauh dari kedua file tersebut, karena ukuran file perbedaannya sangat kecil dan dapat diketahui perbedaan tipe file tidak mempengaruhi performa pada aplikasi FTP dalam jaringan IPv4 murni.

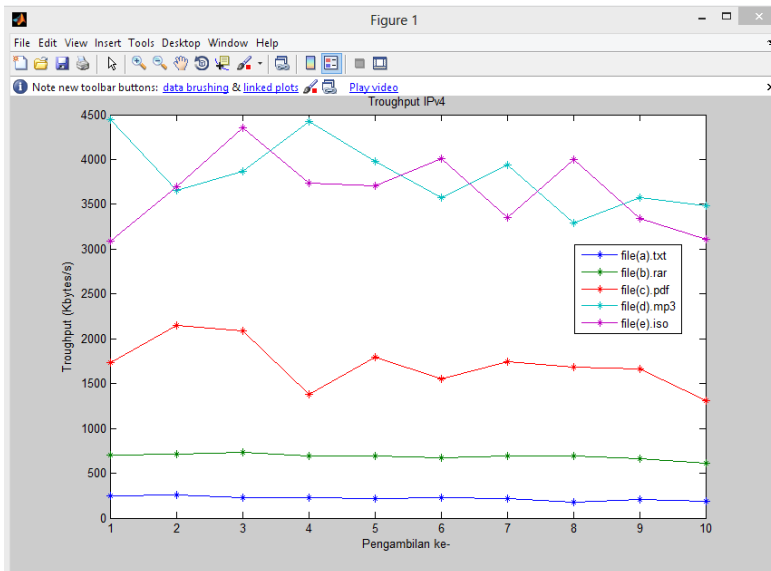
4.4.2 Analisis *Troughput* Jaringan IPv4

Parameter kedua yang diukur pada jaringan IPv4 adalah *troughput*. *Troughput* merupakan kecepatan transfer rata-rata dari suksesnya sebuah paket yang dikirimkan setiap detiknya. Pengambilan dilakukan dengan cara *download* file dari server FTP ke client FTP. Berikut tabel nilai *troughput* yang didapat:

Tabel 4.7 *Troughput* jaringan IPv4 (Kbytes/s)

Pengambilan ke	File (a).txt	File (b).rar	File (c).pdf	File (d).mp3	File (e) .iso
1	252.712121	700.639437	1735.25122	4444.09562	3087.22729
2	257.014234	716.429185	2148.77644	3660.24176	3699.92177
3	228.217443	730.701754	2093.12647	3864.93624	4350.97352
4	225.231152	688.528926	1377.60735	4423.69574	3735.15222
5	219.166895	687.844755	1799.5257	3980.00696	3703.9798
6	228.269924	669.125512	1554.76638	3576.5453	4013.56627
7	217.892252	695.466572	1748.31204	3937.89505	3355.92694
8	175.792345	688.311	1680.98115	3292.77181	4001.98492
9	211.795213	666.955466	1664.43692	3572.62819	3344.33853
10	186.223854	613.226766	1307.96015	3479.69615	3109.71706
Rata-rata	220.231543	685.722937	1711.07438	3823.25128	3640.27883

Agar memudahkan analisis *troughput*, data dari tabel akan ditampilkan dalam bentuk grafik seperti pada gambar 4.19 berikut:



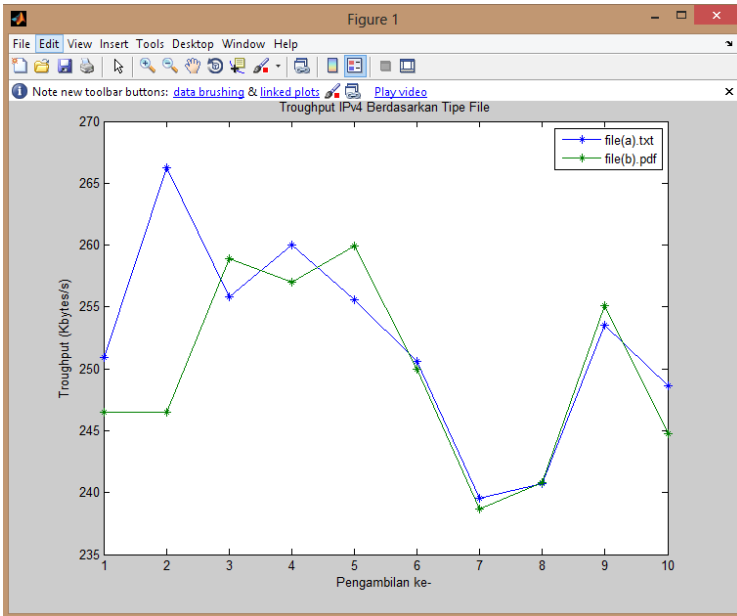
Gambar 4.19 Troughput jaringan IPv4

Berdasarkan tabel 4.7 dan gambar 4.19 dapat dilihat bahwa *troughput* dari file(a).txt memiliki nilai terkecil sebesar 175,7923449 Kbytes/s dan 257,0142342 Kbytes/s untuk yang terbesar. File(b).rar memiliki rentang nilai *troughput* dari 613,2267658 Kbytes/s sampai dengan 730,7017544 Kbytes/s. File ketiga yaitu file(c).pdf mempunyai nilai *troughput* terkecil 1.307,960147 Kbytes/s dan nilai 2.148,776435 Kbytes/s untuk nilai terbesarnya. File selanjutnya yaitu file(d).mp3 mempunyai *troughput* 3.292,771809 Kbytes/s sebagai batas bawah dan 4.444,09562 Kbytes/s sebagai batas paling atasnya. Dan file terakhir memiliki nilai *troughput* terkecil sebesar 3.087,227286 Kbytes/s dan 4.350,973524 sebagai nilai *troughput* terbesar.

Untuk pengambilan data *troughput* pada perbandingan tipe file IPv4 dapat dilihat pada tabel 4.8 dan gambar 4.20.

Tabel 4.8 *Troughput* IPv4 murni berdasarkan tipe file (Kbytes/s)

Pengambilan Ke	File(x).txt	File(y).pdf
1	250.9176692	246.5151856
2	266.2135729	246.5172414
3	255.8164122	258.9348659
4	260.0364384	257.0378788
5	255.613481	259.9153846
6	250.6122777	249.9815157
7	239.5213143	238.70194
8	240.7614693	240.7900356
9	253.5133335	255.0867925
10	248.6135311	244.7765766
Rata-rata	252.1619499	249.8257417



Gambar 4.20 *Troughput* IPv4 berdasarkan tipe file

Dapat dilihat dalam pengambilan data pada file(x).txt memiliki *throughput* terkecil 239,521 Kbytes/s dan *throughput* terbesar pada nilai 266,214 Kbytes/s. Sedangkan untuk file(y).pdf memiliki rentang *throughput* dari 236.702 Kbytes/s sampai dengan 259,915 Kbytes/s. Tidak ada perbedaan *throughput* yang terlalu jauh dari kedua file tersebut, karena ukuran file perbedaannya sangat kecil dan dapat diketahui perbedaan tipe file tidak mempengaruhi performa pada aplikasi FTP dalam jaringan IPv4 murni.

4.5 Analisis Jaringan IPv6

Dalam jaringan Ipv6, parameter yang diukur adalah *latency* dan *throughput* seperti pada jaringan yang lain dalam tugas akhir ini.

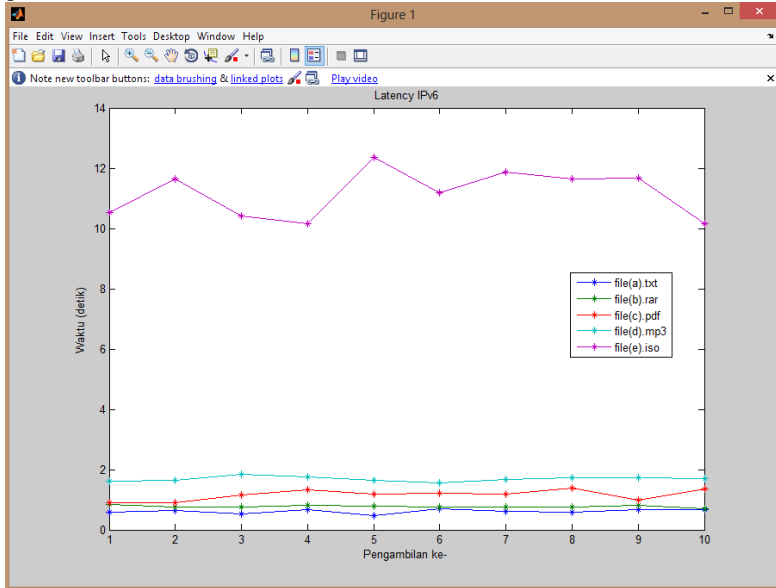
4.5.1 Analisis Latency Jaringan IPv6

Analisis pengolahan data yang pertama adalah *latency* yang dinyatakan dalam satuan sekon atau detik. *Latency* didapat dari waktu *client* melakukan permintaan layanan FTP kepada *server* sampai dengan file yang diminta berada pada *client*. Hasil dari nilai *latency* dapat dilihat pada tabel 4.9.

Tabel 4.9 *Latency* jaringan IPv6 (detik)

Pengambil an ke	File (a).txt	File (b).rar	File (c).pdf	File (d).mp3	File (e) .iso
1	0.572	0.838	0.894	1.608	10.538
2	0.64	0.77	0.903	1.642	11.63
3	0.526	0.77	1.161	1.851	10.41
4	0.661	0.814	1.341	1.747	10.169
5	0.467	0.792	1.176	1.647	12.361
6	0.699	0.755	1.218	1.55	11.194
7	0.624	0.745	1.183	1.68	11.862
8	0.581	0.769	1.378	1.74	11.631
9	0.683	0.801	0.996	1.718	11.672
10	0.671	0.713	1.357	1.699	10.145
Rata-rata	0.6124	0.7767	1.1607	1.6882	11.1612

Data diatas diubah menjadi grafik seperti dapat dilihat pada gambar 4.21.



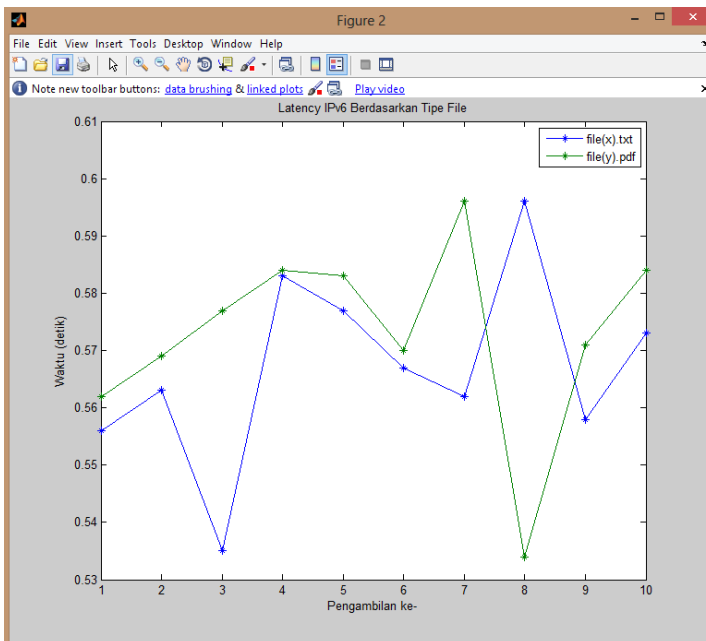
Gambar 4.21 Latency jaringan IPv6

Seperti yang terlihat pada tabel 4.9 dan gambar 4.21, bahwa untuk file(a).txt memiliki nilai *latency* terkecil sebesar 0,467 detik dan terbesar yaitu 0,699 detik. Pada file(b).rar nilai *latency* paling bawah adalah 0,713 detik dan paling atas senilai 0,838 detik. File(c).pdf mempunyai nilai *latency* pada 0,894 detik sampai dengan 1,378 detik. Kemudian pada file(d).mp3 nilai *latency* yang ada adalah 1,55 detik untuk paling bawah dan 1,851 untuk nilai *latency* paling atas. Pada file dengan format iso, nilai *latency* yang ada adalah 10,169 detik dan *latency* tertinggi berada pada nilai 12,361 detik.

Untuk pengambilan data *latency* pada perbandingan tipe file jaringan IPv6 murni dapat dilihat pada tabel 4.10 dan gambar 4.22.

Tabel 4.10 Latency IPv6 murni berdasarkan tipe file (detik)

Pengambilan Ke	File(x).txt	File(y).pdf
1	0.556	0.562
2	0.563	0.569
3	0.535	0.577
4	0.583	0.584
5	0.577	0.583
6	0.567	0.57
7	0.562	0.596
8	0.596	0.534
9	0.558	0.571
10	0.573	0.584
Rata-rata	0.567	0.573



Gambar 4.22 Latency IPv6 berdasarkan tipe file

Dapat dilihat dalam pengambilan data pada file(x).txt memiliki *latency* terkecil 0,535 detik dan *latency* terbesar pada nilai 0,596 detik. Sedangkan untuk file(y).pdf memiliki rentang *latency* dari 0,534 detik sampai dengan 0,596 detik. Tidak ada perbedaan *latency* yang terlalu jauh dari kedua file tersebut, karena ukuran file perbedaannya sangat kecil dan dapat diketahui perbedaan tipe file tidak mempengaruhi performa pada aplikasi FTP dalam jaringan IPv6 murni.

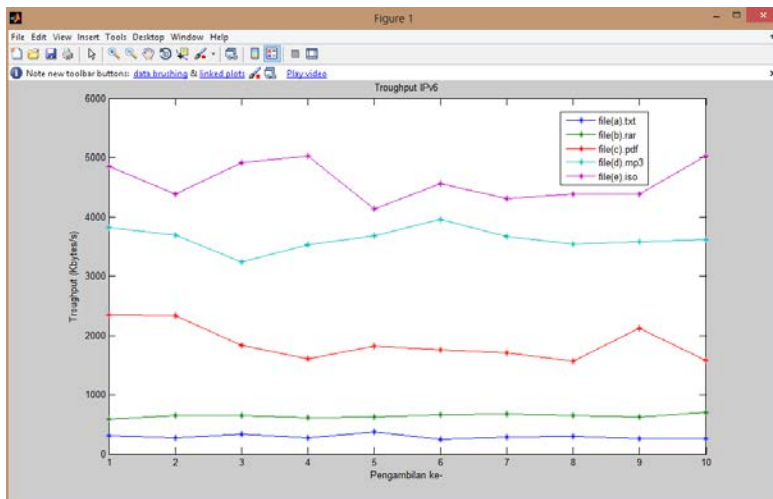
4.5.2 Analisis *Troughput* Jaringan IPv6

Parameter kedua yang diukur pada jaringan IPv6 adalah *troughput*. *Troughput* merupakan kecepatan transfer rata-rata dari suksesnya sebuah paket yang dikirimkan setiap detiknya. Pengambilan dilakukan dengan cara *download* file dari *server* FTP ke *client* FTP. Berikut tabel nilai *troughput* yang didapat.

Tabel 4.11 *Troughput* jaringan IPv6 (Kbytes/s)

Pengambilan ke	File(a).txt	File(b).rar	File(c).pdf	File(d).mp3	File(e) .iso
1	306.5909091	590.9116945	2351.215121	3815.927239	4849.921617
2	270.609375	642.212987	2328.780731	3691.064202	4383.382803
3	329.0912548	645.212987	1833.456176	3241.266904	4906.444957
4	266.3298033	608.039312	1600.818081	3528.22261	5026.582948
5	372.1520343	625.6515152	1811.387204	3680.394189	4129.982607
6	248.6123033	664.7735099	1752.372721	3951.272903	4555.229945
7	277.2916667	673.4281879	1701.258634	3660.954193	4307.592817
8	299.0740103	643.0351105	1560.563514	3541.661552	4383.005674
9	254.2020498	627.7453184	2120.631687	3584.441868	4378.59904
10	262.5692996	699.7545582	1583.176433	3622.254856	5021.450665
Rata-rata	288.6522706	642.0765181	1864.36603	3631.746052	4594.219307

Agar memudahkan dalam menganalisa data, maka data akan disajikan dalam bentuk grafik seperti pada gambar 4.23.



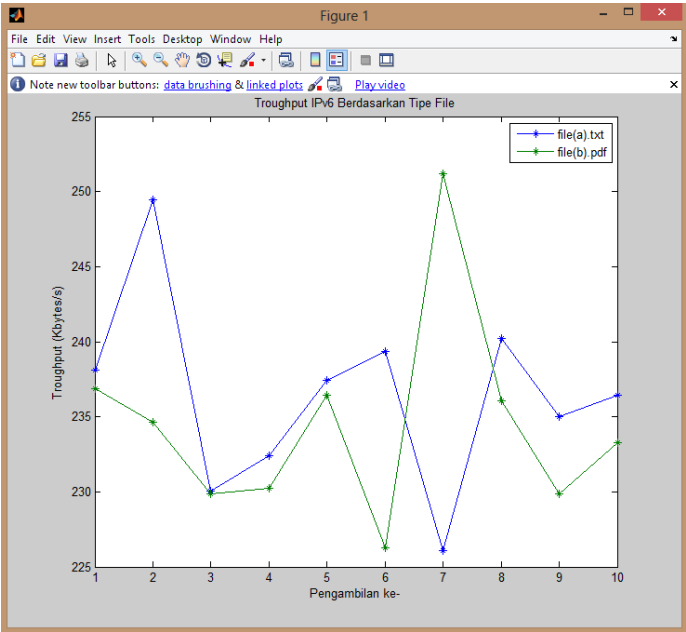
Gambar 4.23 Troughput jaringan IPv6

Dari gambar 4.23 dan tabel 4.11 dapat dilihat bahwa pada file(a).txt memiliki nilai *troughput* terkecil senilai 248,6123 Kbytes/s dan 356,152 Kbytes/s untuk nilai terbesarnya. File(b).rar memiliki rentang antara 590,9117 Kbytes/s sebagai nilai terkecil dan batas atasnya adalah 699,7546 Kbytes/s. Untuk file(c).pdf nilai *troughput*-nya yang paling kecil adalah 1.569,564 Kbytes/s dan nilai terbesar 2.351,215 Kbytes/s. Sedangkan pada file(d).mp3 nilainya adalah 3.241,267 Kbytes/s untuk minimum dan 3.951,273 Kbytes/s untuk maksimum. Dan terakhir pada file dengan ukuran terbesar file(e).iso mempunyai nilai *troughput* dari 4.129,983 Kbytes/s sampai dengan 5.026,583 Kbytes/s. Dari data dapat dilihat bahwa nilai *troughput* berbanding lurus dengan ukuran file.

Untuk pengambilan data *troughput* pada perbandingan tipe file jaringan IPv6 dapat dilihat pada tabel 4.12 dan gambar 4.24.

Tabel 4.12 *Troughput* IPv6 murni berdasarkan tipe file (Kbytes/s)

Pengambilan Ke	Format Txt	Format Pdf
1	238.1172291	236.8769772
2	249.4224299	234.6343154
3	230.051458	229.869863
4	232.4124783	230.2607204
5	237.4585538	236.4666667
6	239.3355872	226.2818792
7	226.0989933	251.2059925
8	240.2240143	236.0577933
9	235.0139616	229.869863
10	236.4585538	233.2443281
Rata-rata	236.4593259	234.4768399



Gambar 4.24 *Troughput* IPv6 berdasarkan tipe file

Dapat dilihat dalam pengambilan data pada file(x).txt memiliki *throughput* terkecil 226,099 Kbytes/s dan *throughput* terbesar pada nilai 249,422 Kbytes/s. Sedangkan untuk file(y).pdf memiliki rentang *throughput* dari 226.282 Kbytes/s sampai dengan 251,205 Kbytes/s. Tidak ada perbedaan *throughput* yang terlalu jauh dari kedua file tersebut, karena ukuran file perbedaannya sangat kecil dan dapat diketahui perbedaan tipe file tidak mempengaruhi performa pada aplikasi FTP dalam jaringan IPv6 murni.

4.6 Analisis Perbandingan Parameter

Analisis perbandingan parameter akan menunjukkan masing-masing nilai *latency* dan *throughput* antar jaringan. Perbandingan tersebut dapat terlihat performa jaringan yang lebih baik diantara IPv6 *tunnel broker*, IPv4 dan IPv6 pada transfer file dengan menggunakan FTP. Data yang disajikan adalah rata-rata dari parameter yang ada.

4.6.1 Analisis Perbandingan *Latency* Ukuran Berbeda

Perbandingan nilai *latency* dari semua jaringan yang dilakukan pada tugas akhir ini akan diambil rata-ratanya untuk dianalisa. *Latency* dari lima buah file tersebut akan dilihat perbandingan persentasenya pada masing-masing jaringan. Berikut rata-rata *latency* yang disajikan pada tabel 4.13.

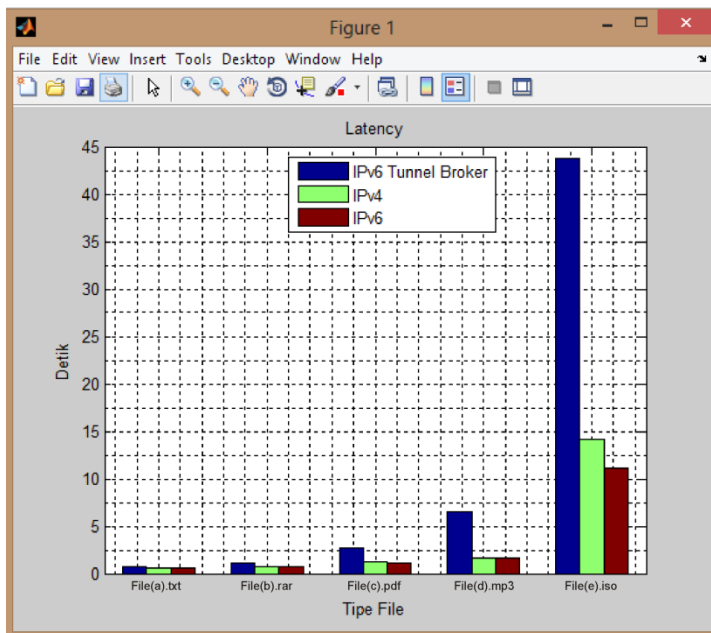
Tabel 4.13 Rata-rata *latency* (detik)

Jenis File	IPv6 <i>Tunnel Broker</i>	IPv4	IPv6
File(a).txt	0.6977	0.6356	0.6124
File(b).rar	1.1242	0.7236	0.7767
File(c).pdf	2.7611	1.2764	1.1607
File(d).mp3	6.5067	1.6273	1.6882
File(e).iso	43.7095	14.0881	11.1612

Agar memudahkan dalam menganalisa data, maka data akan disajikan dalam bentuk diagram seperti pada gambar 4.25 dan perbandingan persentase performa *latency* pada masing-masing jaringan di setiap pengiriman file nya dapat dilihat pada tabel 4.14.

Tabel 4.14 Persentase *latency* antar jaringan

Jenis File	IPv6 Tunnel Broker v IPv4 (%)	IPv6 Tunnel Broker v IPv6 (%)	IPv4 v IPv6 (%)	IPv6 v IPv4 (%)
File (a).txt	9.770295784	13.9288047	3.7883736	-3.65009
File (b).rar	55.3620785	44.74056907	-6.8366165	7.338308
File (c).pdf	116.3193356	137.8823124	9.9681227	-9.06456
File (d).mp3	299.8463713	285.4223433	-3.6073925	3.742395
File (e) .iso	210.2583031	291.6200767	26.223883	-20.7757



Gambar 4.25 Rata-rata *latency*

Pada gambar 4.25 terlihat perbandingan masing-masing *latency* pada tipe file selama pengiriman melalui FTP. Dari kelima file tersebut, *latency* yang dimiliki oleh jaringan IPv6 *tunnel broker* selalu lebih lambat dibandingkan dengan jaringan yang lain. Pada tipe file(a).txt dengan format ASCII, jaringan IPv6 *tunnel broker* lebih besar 9,77% bila dibandingkan dengan jaringan IPv4 dan 13,93% dari jaringan IPv6. Untuk tipe file berformat *binary*, performa jaringan IPv6 *tunnel broker* bila dibandingkan dengan jaringan IPv4 untuk file(b).rar lebih besar 55,36%, file(c).pdf 116,31%, file(d).mp3 299,84%, dan file berukuran terbesar file(e).iso 210,26%. Perbandingan antara jaringan IPv6 *tunnel broker* dan jaringan IPv6 pada tipe file yang memiliki format *binary*, nilai *latency tunnel broker* lebih besar file(b).rar 44,74%, file(c).pdf 137,88%, file(d).mp3 285,42% dan file(e).iso senilai 291,62%.

Sedangkan untuk perbandingan jaringan IPv4 dan IPv6, untuk IPv4 lebih besar dalam *latency* pada file(a).txt, file(c).pdf, dan file(e).iso dengan persentase 3,79%, 9,97%, dan 26,22% dibandingkan dengan jaringan IPv6 atau jaringan IPv6 lebih kecil 3,65%, 9,06%, dan 20,77%. Pada file(b).rar dan file(d).mp3 *latency* jaringan IPv6 lebih besar senilai 7,33% dan 3,74% dibandingkan dengan IPv4 atau IPv4 lebih kecil nilai *latency* nya 6,84% dan 3,06% daripada IPv6.

4.6.2 Analisis Perbandingan *Troughput* Ukuran Berbeda

Analisa *troughput* juga sama dengan analisa sebelumnya. *Troughput* rata-rata dari setiap pengambilan file akan dibandingkan pada masing-masing jaringan. Kemudian dilihat persentase perbandingan *troughput* antar jaringan. Berikut rata-rata nilai *troughput* pada tabel 4.15.

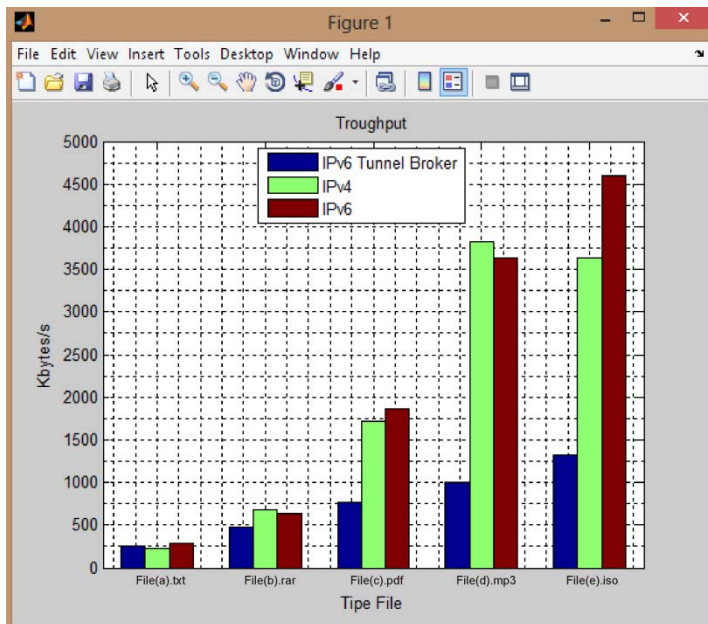
Tabel 4.15 Rata-rata *troughput* (Kbytes/s)

Jenis File	IPv6 Tunnel Broker	IPv4	IPv6
File(a).txt	249.7613334	220.2315433	288.6522706
File(b).rar	470.990749	685.7229371	642.0765181
File(c).pdf	763.1254666	1711.07438	1864.36603
File(d).mp3	1008.59634	3823.251282	3631.746052
File(e).iso	1319.495436	3640.278832	4594.219307

Agar memudahkan dalam menganalisa data, maka data akan disajikan dalam bentuk diagram seperti pada gambar 4.26 dan perbandingan persentase performa *troughput* pada masing-masing jaringan di setiap pengiriman file nya dapat dilihat pada tabel 4.16.

Tabel 4.16 Persentase *troughput* antar jaringan

Jenis File	IPv6 Tunnel Broker v IPv4 (%)	IPv6 Tunnel Broker v IPv6 (%)	IPv4 v IPv6 (%)	IPv6 v IPv4 (%)
File (a).txt	13.40851982	-13.473283	-23.7035126	31.06763287
File (b).rar	-31.3147157	-26.6456979	6.797697444	-6.36502247
File (c).pdf	-55.4008011	-59.0678303	-8.22218643	8.958795276
File (d).mp3	-73.6194075	-72.2283352	5.2730898	-5.0089627
File (e) .iso	-63.752902	-71.279224	-20.7639299	26.20514855



Gambar 4.26 Rata-rata *troughput*

Seperti terlihat pada gambar 4.26 pada setiap perubahan ukuran file juga terjadi peningkatan nilai *throughput*. Untuk perbandingannya cukup relatif pada masing-masing jaringan. Pada file(a).txt nilai *throughput* selisih antar jaringan tidak terlalu besar, nilai *throughput* jaringan IPv4 lebih kecil 13,41% dibandingkan jaringan IPv6 *tunnel broker* dan 23,7% dengan IPv6. Sedangkan untuk IPv6 *tunnel broker* lebih kecil sebesar 13,47% dibanding dengan IPv6. Untuk file dengan mode *binary*, IPv6 *tunnel broker* memiliki nilai *throughput* lebih kecil 31,31% , 55,4%, 73,62%, dan 63,75% dibandingkan dengan jaringan IPv4 tiap-tiap filenya. Sedangkan *throughput* perbandingan IPv6 *tunnel broker* dengan IPv6 akan lebih kecil sebesar 26,64%, 59,07%, 72,23%, dan 71,28% dibandingkan pada jaringan IPv6 pada setiap tipe file *binary*.

Untuk perbandingan antara IPv4 dan IPv6 cukup bervariasi. Tiga buah file yaitu file(a).txt, file(c).pdf, dan file(e).iso pada IPv6 lebih besar dibandingkan dengan IPv4 yaitu sebanyak 31,06%, 8,96%, dan 26,21% atau nilai *throughput* IPv4 lebih kecil 23,7%, 8,22%, dan 20,76% dibandingkan dengan IPv6. Sedangkan untuk nilai *throughput* IPv4 yang lebih besar dibandingkan dengan IPv6 ada pada *throughput* file(b).rar dan file(d).mp3 yaitu sebesar 6,79% dan 5,27% yang berarti IPv6 mempunyai nilai *throughput* lebih kecil 6,37% dan 5,01% pada dua file tersebut dibandingkan dengan IPv4.

4.7 Analisis Keseluruhan

Pada hasil yang ada, jenis atau tipe file tidak mempunyai pengaruh besar terhadap parameter yang diukur (*latency* dan *throughput*). Tetapi yang memiliki pengaruh besar adalah ukuran dari file itu sendiri. Pada tipe mode *binary* aplikasi *file transfer protocol* akan mengirimkan file dengan cara bit per bit di sisi pengirim, sedangkan penerima akan menerimanya dalam cara aliran bit (*bitstream*). Sedangkan pada mode ASCII akan dianggap sebagai format teks berformat ASCII, pihak yang menerima akan bertanggung jawab untuk menerjemahkan format teks yang diterima ke salah satu yang kompatibel dengan sistem operasi yang ada. Selain itu untuk tipe file yang berukuran besar, nilai dari *latency* dan *throughput*-nya cenderung tidak stabil atau memiliki rentang perubahan yang cukup besar.

Nilai *latency* akan berbanding terbalik dengan nilai *throughput*. Jika semakin besar nilai *latency*, maka menyebabkan peningkatan nilai

throughput. Jika semakin kecil nilai *latency*, maka akan semakin besar pulalah nilai dari *throughput*. Hal ini menunjukkan bahwa adanya hubungan antara *throughput* dan *latency* pada satu file yang sama. Nilai *throughput* didapatkan dari pembagian ukuran file yang dikirimkan oleh server dengan *latency* selama pengiriman file, jadi semakin besar *latency* semakin kecil nilai *throughput*, begitupun sebaliknya.

Pada transfer file yang melewati jaringan IPv6 *tunnel broker*, akan dilakukan proses enkapsulasi dan dekapsulasi paket FTP yang dikirimkan. Selain itu pada server FTP akan melakukan *extension* yang memungkinkan sebuah FTP server melakukan transfer data anatara node-node jaringan dengan tambahan fitur keamanan atau yang disebut dengan mode *extended passive mode*. Pada *extended passive mode* akan melakukan perintah pengecekan setiap paket walaupun dalam status transfer data yang membuat waktu transfer akan menjadi sedikit lambat (memperbesar nilai *latency*).

4.8 Strategi Implementasi *Tunnel Broker*

IPv6 *tunnel broker* adalah salah satu metode transisi IPv4 ke IPv6 yang dapat dimanfaatkan ketika sebuah perangkat komputer ingin mendapatkan akses ke server namun terisolasi oleh jaringan yang berbeda. Dengan metode *tunnel broker* hal itu dapat diatasi dengan membuat kanal yang akan menembus jaringan tersebut agar paket data bisa sampai ke server sehingga bisa diakses oleh pengguna. Maka dari itu dirasa perlu dilakukan perencanaan strategi untuk pengimplementasian IPv6 *tunnel broker* jika memang dibutuhkan dalam menunjang penelitian dan pembelajaran di kampus ITS. Berikut strategi yang bisa diterapkan:

1. Melakukan persiapan kemampuan kampus terhadap daya dukung IPv6 seperti perangkat yang akan digunakan dan *internet service provider* yang bekerjasama dengan ITS sudah mendukung IPv6. Sampai tahapan ini, ITS masih sedang dalam proses menuju kemampuan dalam menunjang IPv6, namun kemampuan tersebut hanya sebatas pada tataran atas (pada sisi *core network*).
2. Harus dilakukan survey terhadap penggunaan perangkat router yang banyak digunakan oleh kampus ITS. Berdasarkan keadaan saat ini, sebagian besar kampus ITS menggunakan perangkat milik cisco, windows, dan linux. Mengetahui hal ini cukup penting dikarenakan pada saat pengguna mengakses *tunnel broker*, akan mendapatkan konfigurasi router sesuai dengan tipe yang dimilikinya.

3. Membuat *database* tentang pengguna. Melalui *database*, *tunnel broker* bisa dibatasi dan dikontrol penggunaannya karena *user* sebelum mendapatkan akses untuk membuat, memodifikasi, ataupun menghapus *tunnel* harus melakukan registrasi terlebih dahulu. Isi dari *database* tersebut yang harus dimasukkan adalah nama, nama lengkap, NRP, nama pengguna, kata sandi, verifikasi kata sandi, alamat asal, alamat Surabaya, email ITS, email lainnya, nomor telepon. Beberapa hal yang cukup penting adalah NRP, nama pengguna, kata sandi, dan email ITS dikarenakan elemen tersebut akan diperlukan ketika *user* melakukan *login* ke *tunnel broker* dan diatur bahwa penggunaan *tunnel broker* hanya orang-orang yang memiliki alamat email ITS (warga ITS).
4. Membuat *database* ketersediaan alamat IPv6. Pada *database* ini akan berisi tentang alamat-alamat IPv6 yang akan diberikan kepada *user*. Selain itu perlu juga dimasukkan pengaturan *timeout*, yaitu masa waktu penggunaan alamat IPv6 agar tidak ada alamat IPv6 yang sedang berada pada *user* namun *user* tersebut sebenarnya sedang tidak aktif.
5. Membangun perangkat *tunnel broker*. *Tunnel broker* adalah wadah koneksi dari *user* pada jaringan IPv4 untuk melakukan proses pendaftaran dan pengaktifan *tunnel*. Fungsinya adalah untuk mengatur pembentukan, modifikasi, dan penghapusan *tunnel* sesuai dengan permintaan dari *user*. Dalam prakteknya, *tunnel broker* bertanggung jawab untuk membagi beban jaringan kepada *tunnel server*, caranya adalah dengan mengirimkan konfigurasi kepada *tunnel server* yang bersangkutan pada saat *tunnel* tersebut dibentuk, dimodifikasi, atau dibubarkan. *Tunnel broker* juga harus mendaftarkan alamat IPv6 *user* dan memasukkannya ke dalam DNS *server*. *Tunnel broker* haruslah mendukung IPv4 namun tidak harus mendukung IPv6, karena *tunnel broker* akan berhubungan secara langsung dengan internet melalui jaringan IPv4 dan hubungan antara *tunnel broker* dan *tunnel server* dapat berupa IPv6 maupun IPv4. Selain itu *tunnel broker* dapat dilengkapi dengan otentifikasi, otorisasi dan akuntansi untuk manajemen *user* dan akuntansi *tunnel*. Yang dibutuhkan hanyalah sebuah server yang berfungsi sebagai *tunnel broker* serta memasukkan template konfigurasi router untuk pengguna.
6. Membangun perangkat DNS *server*. DNS *server* berfungsi sebagai penerjemah nama domain menjadi alamat IP sehingga misalkan

- nama domain yang akan diakses oleh *user* yaitu *www.tunnelbroker.its.ac.id* ketika diakses akan diterjemahkan kedalam bentuk IP. DNS server bisa berupa perangkat yang sama dengan *tunnel broker* (digabung).
7. Membangun perangkat *tunnel server*. *Tunnel server* adalah perangkat router *dualstack* yang terhubung dengan *tunnel broker* dan *interface* yang lain terhubung dengan jaringan IPv6 pada ISP. *Tunnel server* bisa digabung dengan *tunnel broker*, namun apabila penggunaannya untuk umum seperti pada kampus sebaiknya dibuatkan perangkat tersendiri agar tidak terlalu banyak terbebani ketika ada banyak *user* yang mengakses secara bersamaan. Selain itu, tidak hanya dibutuhkan satu buah *tunnel server*, namun harus beberapa *tunnel server* agar ada alternatif dari *user* akan menggunakan *tunnel server* yang secara otomatis akan diatur oleh *tunnel broker* dan jumlahnya juga disesuaikan dengan prediksi pengguna IPv6 *tunnel broker*.
 8. Menambahkan keamanan pada jaringan berupa firewall dan proxy agar bisa dibatasi penggunaan dalam lingkup kampus dan hanya bisa digunakan oleh civitas akademi kampus.
 9. Melakukan pelatihan ataupun *workshop* bagi calon pengguna di kampus seperti para dosen, warga laboratorium, penduduk di dalam kampus. Hal ini bertujuan selain untuk mengajarkan tentang penggunaan *tunnel broker* juga untuk melakukan sosialisasi IPv6 *tunnel broker*.

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

Setelah dilakukan pengambilan data dan melakukan analisis terhadap penelitian, maka dapat ditarik kesimpulan. Selain itu disertakan juga terkait dengan kendala dan saran-saran terhadap penelitian ini yang bisa digunakan untuk pengembangan dan kelanjutan diwaktu yang akan datang.

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini setelah dilakukan pengukuran dan analisis data adalah:

1. Untuk file yang berukuran kecil yaitu file(a).txt sebesar 166,790 KB *latency* pada jaringan IPv6 *tunnel broker* akan lebih besar 9,77% dibanding jaringan IPv4 dan 13,93% dibanding dengan IPv6. Sedangkan *throughput* dari IPv6 *tunnel broker* lebih besar 13.4% terhadap IPv4 dan lebih kecil 13,47% dari IPv6.
2. Untuk file yang berukuran besar yaitu file(e).iso sebesar 51.013,632 KB *latency* pada jaringan IPv6 *tunnel broker* akan lebih besar 210,26% dibanding jaringan IPv4 dan 291,6% dibanding dengan IPv6. Sedangkan *throughput* dari IPv6 *tunnel broker* lebih kecil 63.75% terhadap IPv4 dan 71,27% dari IPv6.
3. Perlu dilakukan pengecekan terhadap IOS router terhadap daya dukung kemampuan pengalamatan IPv6 dan pembentukan *tunnel* dengan cara memeriksa versi dan flash dari router tersebut.
4. Pada metode *tunnel broker*, akan terjadi proses enkapsulasi paket IPv4 sehingga paket IPv6 akan diperlakukan seperti paket IPv4 yang lain. Sehingga paket tersebut dengan mudah dapat dikirimkan melalui jaringan IPv4. Namun akan ada penambahan IP *header* dan keamanan pada paket FTP yang secara tidak langsung akan berpengaruh terhadap performansi jaringan.
5. Nilai *throughput* akan berbanding terbalik dengan nilai *latency*. Karena *throughput* didapat dari ukuran file dibagi nilai *latency* serta adanya penambahan keamanan pada paket FTP dan proses enkapsulasi dekapsulasi paket IPv6.
6. Perbedaan tipe file dalam pengiriman yang berbasis aplikasi FTP tidak mempengaruhi kinerja nilai parameter (*latency* dan *throughput*) secara signifikan. Ada perbedaan dalam pengiriman pada mode

ASCII dan binary, namun perbedaannya tidak seberapa besar. Yang lebih berpengaruh adalah ukuran file. Bahkan untuk ukuran file yang besar, ada perbedaan *latency* dan *troughput* yang jauh pada setiap pengambilan datanya.

5.2 Saran

Saran untuk pengembangan analisis performa IPv6 *tunnel broker* antara lain:

1. Melakukan perbandingan kinerja dengan protokol selain *file transfer protocol* dan metode transisi IPv4 dan IPv6 yang lain.
2. Pembentukan konfigurasi *tunnel* diperbanyak terhadap *operating system* yang ada dan ditambahkan pada *database*. Sehingga tidak terpaku pada hanya satu jenis perangkat router.
3. Kedepannya untuk pengujian mekanisme *tunnel broker* bisa menggunakan jaringan yang sebenarnya atau diberikan *traffic* pada jaringan.

DAFTAR PUSTAKA

- [1]. Sugeng Winarno, "*Jaringan Komputer dengan TCP/IP*", 2010.
- [2]. Damon Reed, "*Practical Assignment version 1.4b Option One, Applying the OSI Seven Layer Network Model To Information Security*", 2003.
- [3]. J. Postel, "File Transfer Protocol," 1985.
- [4]. A. Durand, "'Deploying IPv6'," *IEEE Internet Computing*, pp. pp. 79-81, Jan-Feb 2001.
- [5]. A.K. Yeo and A.L. Ananda K. Wang, "*DTTS: a Transparent and Scalable Solution for IPv4 to IPv6 Transition*", Proceedings Of The Tenth International Conference on Computer Communications and Networks, pp. pp.248-253, 2001.
- [6]. Onno W. Purbo, "*TCP/IP*", Elex Media Komputindo, 2000.
- [7]. "*A Beginner's Guide FTP 101*", 2006.
- [8]. Reko Ardonto, "*Analisa dan Implementasi Ipv 6 Tunnel Broker untuk Interkoneksi antara IPv6 dan IPv4*", Universitas Diponegoro.
- [9]. Aris Cahyadi Risdianto, "*IPV6 Tunnel Broker Implementation and Analysis for IPv6 and IPv4 Interconnection*", The 6th International Conference on Telecommunication Systems, Services, and Applications, 2011.
- [10]. T. Dunn, "'The IPv6 Transition", *IEEE Internet Computing*, Vol.6. No.3, May/June 2002, pp. pp.11-13.
- [11]. Amoss J. Jonas dan Minoli Daniel, "*Handbook of IPv4 to IPv6 Transition Metodologies for Institutional and Corporate Networks*", Auerbach Publications, 2008.
- [12]. Ettikan Kandasamy, "*Application Performance Analysis in Transition Mechanism from IPv4 to IPv6*", Research & Business Development Department, Malaysia.
- [13]. RFC 3053, "*IPv6 Tunnel Broker*", Internet Society, 2001.

Halaman ini sengaja dikosongkan

RIWAYAT HIDUP



Muhammad Yusro Muhtadi dilahirkan di kota Banjarbaru pada tanggal 02 Desember 1991 dari ayah yang bernama M. Yusni dan ibu bernama Rochmiyati. Penulis merupakan anak sulung dari dua bersaudara. Penulis menyelesaikan pendidikan di Sekolah Dasar Negeri Banjarbaru Utara 1 pada tahun 2004. Kemudian Penulis melanjutkan pendidikan di SMP Negeri 1 Banjarbaru dan selesai pada tahun 2007. Untuk tingkat SMA, penulis berhasil menyelesaikannya di SMA Negeri 1 Banjarbaru pada tahun 2010. Pada tahun yang sama, penulis hijrah ke Surabaya dan diterima sebagai mahasiswa Teknik Elektro Institut Teknologi Sepuluh Nopember dan selesai pada Maret 2015. Selama kuliah Penulis aktif di organisasi kemahasiswaan seperti BEM ITS dan HIMATEKTRO ITS sebagai sekretaris umum. Penulis juga aktif di dunia kepanduan dari tahun 2011 sampai 2014. Selain itu, Penulis juga menjadi asisten di Laboratorium Telekomunikasi Multimedia.

Halaman ini sengaja dikosongkan

Lampiran A

Jurusan Teknik Elektro
Fakultas Teknologi Industri – ITS

TE 091399 TUGAS AKHIR – 4 SKS

Nama Mahasiswa : M. Yuro Muhtadi
Nomer Pokok : 2210100155
Bidang Studi : Telekomunikasi Multimedia
Tugas Diberikan : Semester Genap Th. 2013/2014
Dosen Pembimbing : 1. Dr. Ir. Achmad Affandi, DEA
2. Ir. Djoko Suprajitno Rahardjo, MT.
Judul Tugas Akhir : **Analisis Unjuk Kerja Interkoneksi IPv6 dan IPv4 dengan Metode IPv6 Tunnel Broker pada Kampus ITS Surabaya**
(*Performance Analysis IPv6-IPv4 Interconnection Using IPv6 Tunnel Broker Method On ITS Network*)

28 FEB 2014

Uraian Tugas Akhir :

Sebagai salah satu kampus yang berbasis teknologi, seharusnya kampus Institut Teknologi Sepuluh Nopember Surabaya (ITS Surabaya) mampu menerapkan pemanfaatan teknologi terbaru salah satunya adalah IP versi 6 untuk menunjang aktivitas akademik yang ada. IP versi 6 (IPv6) merupakan protokol internet baru yang dikembangkan untuk menggantikan IP versi 4 (IPv4) yang saat ini tengah mendekati ambang batas alokasi alamatnya. Ruang alamat IPv6 ini menggunakan sistem pengalamatan 128 bits yang berarti mampu mengalokasikan alamat IP sebanyak 296 kali lebih banyak dibandingkan IPv4. Penyebaran IPv6 dalam menggantikan IPv4 memakan waktu yang sangat lama sehingga pada masa ini akan tercipta kondisi jaringan Internet di mana IPv6 dan IPv4 berjalan bersamaan. Dengan demikian, diperlukan mekanisme transisi untuk menjembatani keduanya agar dapat saling berkomunikasi salah satunya dengan metode IPv6 *Tunnel Broker*.

Dalam Tugas Akhir ini akan dibangun suatu jaringan yang menggambarkan kondisi jaringan IPv6 yang dikoneksikan dengan jaringan IPv4 dengan menggunakan sistem IPv6 *Tunnel Broker* di kampus ITS. Hasil yang didapat menunjukkan bagaimana cara kerja dan dilakukan analisa pengujian terhadap sistem IPv6 *Tunnel Broker* dengan melakukan pengiriman data dimana IPv6 akan berlaku sebagai *server* dan IPv4 sebagai *client* atau sebaliknya. Pengujian dilakukan untuk mengetahui performansi *Quality of Service* (QoS) dengan menggunakan parameter *delay*, *jitter*, *packet loss*, dan *throughput*. Pada tugas akhir ini juga diteliti tentang performansi *server* dengan melihat *CPU Utilization*.

Kata Kunci : IPv6, IPv4, IPv6 *Tunnel Broker*, *Quality of Service*

Dosen Pembimbing I,

Dosen Pembimbing II,

Dr. Ir. Achmad Affandi, DEA
NIP 1965 10 14 1990 02 1001

Ir. Djoko Suprajitno Rahardjo, MT.
NIP 1955 06 22 1987 01 1001

Mengetahui,
Jurusan Teknik Elektro FTI – ITS
Ketua

Dr. Tri Arief Sardiono, ST, MT.
NIP. 1970 02 12 1995 12 1001

Menyetujui,
Bidang Studi Telekomunikasi Multimedia
Koordinator,

Dr. Ir. Endrovono, DEA
NIP. 1965 04 04 1991 02 1001

A. JUDUL TUGAS AKHIR

Analisis Unjuk Kerja Interkoneksi IPv6 dan IPv4 dengan metode IPv6 *Tunnel Broker*

B. RUANG LINGKUP

- Interkoneksi IPv6 dan IPv4
- *IPv6 Tunnel Broker*
- *latency* dan *troughput*

C. LATAR BELAKANG

Internet yang berkembang cukup pesat telah memunculkan generasi baru *internet protocol* yang disebut dengan IPv6. IPv6 merupakan sebuah protokol yang telah dirancang oleh IETF (Internet Engineering Task Force) untuk menggantikan IPv4. IPv6 memiliki kapasitas *address* 340 *undencillion* alamat publik, penyusunan alamat lebih terstruktur yang memungkinkan internet untuk terus berkembang dan menyediakan *routing* baru yang tidak terdapat pada IPv4. Jumlah yang sangat besar ini dapat menyelesaikan permasalahan dari penggunaan IPv4 mengingat semakin banyaknya pengguna internet yang berdampak pada kapasitas penggunaan melalui IPv4 penuh sesak dan berada pada ambang batasnya. IPv6 dilengkapi sebuah mekanisme penggunaan *address* secara lokal yang memungkinkan terwujudnya instalasi secara *Plug & Play*, serta dukungan terhadap aliran data secara *realtime*, *mobilitas host*, *end-to-end security*, ataupun konfigurasi otomatis. Selain itu IPv6 memiliki tipe *address anycast* yang dapat digunakan untuk pemilihan *route* secara efisien. Dengan berbagai keunggulan yang ada perlu dilakukan transisi dari IPv4 menuju IPv6 melalui proses metode transisi sesuai kebutuhan.

Metode *Dual IP Stack*, *Address Protocol Translator* dan *Tunneling* dapat digunakan untuk transisi IPv4 menuju IPv6. *Dual IP Stack* digunakan agar perangkat bisa menggunakan IPv4 dan IPv6 secara bersama-sama, *Address Protocol Translator* mengijinkan *node* IPv6 murni untuk berhubungan dengan *node* IPv4 murni dengan memanfaatkan protokol translasi alamat, dan *tunneling* memungkinkan pengiriman *traffic* IPv6 melalui infrastruktur jaringan IPv4.

Kampus Institut Teknologi (ITS) Surabaya sebagai kampus yang berbasiskan keteknikan harusnya mampu menerapkan penggunaan teknologi terkini salah satunya dalam pemanfaatan IPv6 untuk jaringan

internet kampus. Metode yang cocok digunakan dalam mendukung interkoneksi IPv4 dan IPv6 adalah sistem *IPv6 tunnel broker* dimana *tunnel* diaktifkan secara otomatis oleh *tunnel broker* kepada *dual stack host* IPv6/IPv4 yang terisolasi dengan jaringan IPv6 yang lain agar bisa berhubungan dengan jaringan IPv6 tersebut, melalui jaringan IPv4 yang sudah ada. Selain itu juga akan dilihat perbandingan penggunaan IPv4 dan IPv6 dalam jaringan kampus dengan parameter yang ada.

D. PERUMUSAN MASALAH

Masalah yang akan dibahas dalam Tugas Akhir ini adalah :

1. Bagaimanakah pengimplementasian interkoneksi Ipv6 dan IPv4 dengan sistem IPv6 Tunnel Broker?
2. Bagaimanakah performa *download file* dari hasil penerapan sistem IPv6 Tunnel Broker pada kampus ITS?

E. TUJUAN TUGAS AKHIR DAN MANFAAT

Penelitian pada tugas akhir ini bertujuan sebagai berikut :

1. Membuat sistem *IPv6 Tunnel Broker*
2. Melihat unjuk kerja dari interkoneksi IPv6 dan IPv4 melalui sistem *IPv6 Tunnel Broker*.
3. Melihat pengaruh *download file* dari *client* ke *server* melalui sistem *IPv6 Tunnel Broker*.

Adapun manfaat yang ingin didapatkan dari Tugas Akhir ini adalah:

1. Adanya sistem jaringan internet yang lebih baik di kampus ITS
2. Berkembangnya teknologi jaringan terbaru untuk kampus ITS.

F. TINJAUAN PUSTAKA DAN DASAR TEORI

• TCP dan IP

TCP dan IP merupakan salah satu standar protokol yang dirancang untuk melakukan fungsi-fungsi komunikasi data dalam jaringan internet. TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dalam komunikasi data. Dengan prinsip ini maka tugas masing-masing protokol menjadi jelas dan sederhana, sehingga mudah untuk diimplementasikan di seluruh perangkat dan perangkat lunak jaringan dan juga mudah dalam melakukan proses *trouble shooting*. Karena beberapa kelebihan yang

dimiliki protokol TCP/IP ini, maka saat ini TCP/IP lebih banyak digunakan dengan standar protokol yang lain

Arsitektur TCP/IP dapat dimodelkan dalam empat lapisan TCP/IP, yaitu *network interface layer*, *network layer*, dan *application layer*.

<i>Application layer</i>
<i>Transport layer</i>
<i>Network layer</i>
<i>Network Interface layer</i>

Gambar 1. Arsitektur Protokol TCP/IP

Dalam proses pengiriman data antar layer, setiap layer akan menganggap informasi yang datang dari layer sebelumnya sebagai data, sehingga ia akan menambahkan informasi miliknya pada data tersebut. Begitu juga sebaliknya, jika ia menerima data yang dianggap valid maka ia akan melepas informasi tersebut.

Network Interface Layer merupakan lapisan terbawah yang bertanggung jawab untuk mengirim dan menerima data ke dan dari media fisik. Oleh karena protokol dalam layer ini harus mampu merubah bit-bit informasi menjadi sinyal listrik. Contoh dari protokol dalam layer ini adalah PPP, SLIP dan Ethernet. PPP (*Point to Point Protocol*) adalah protokol yang biasa dipakai pada komunikasi *router to router* dan *host to network* diatas jaringan *asynchronous* dan *synchronous*. SLIP (*Serial Line in Protocol*) adalah protokol sebelum PPP dimana teknik enkapsulasinya lebih sederhana dari PPP. Ethernet adalah standard IEEE 802.3 untuk komunikasi dua komputer atau lebih, Ethernet menggunakan CSMA/CD (*Carrier Sense Multiple Access/Collision Detection*) yaitu metode agar tidak saling mengirimkan informasi secara bersamaan. Setiap *ethernet card* mempunyai 48 bit sebagai alamatnya.

Internet Layer merupakan protokol yang bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat dan bersifat *unreliable* dan *connectionless*. Pada layer ini terdapat tiga macam protokol yaitu IP, ARP dan ICMP. Internet protokol berfungsi untuk menyampaikan paket data ke alamat yang tepat. ARP (*Address Resolution Protocol*) ialah protokol yang digunakan untuk menemukan alamat *hardware* dari LAN *card*.

Transport Layer merupakan protokol yang bertugas untuk mengadakan hubungan dan mengatur transportasi data antara dua

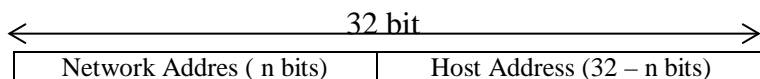
host/komputer. Protokol dalam lapisan ini, yaitu TCP dan UDP. TCP (*Transmission Control Protocol*) bersifat *reliable* dan *connection oriented*, Sedangkan UDP (*Unit Datagram Protocol*) bersifat *connectionless* dan *unreliable*.

Application Layer, merupakan lapisan teratas yang berisi semua aplikasi berbasis TCP & IP dan berhubungan langsung dengan pemakai. Aplikasi tersebut misalnya FTP, HTTP dan Telnet . FTP (*File Transfer Protokol*) adalah program aplikasi untuk mentransfer file antara *clien* & *server*. HTTP (*Hyper Text Transfer Protokol*) adalah program aplikasi yang digunakan untuk menterjemahkan alamat IP menjadi susunan huruf yang dipisahkan dengan tanda “.” misal <http://www.its.ac.id>. Dari beberapa macam protokol yang ada dalam TCP & IP, protokol IP merupakan inti dari protokol TCP & IP. Seluruh data yang berasal dari lapisan diatas IP harus dilewatkan, diolah oleh protokol IP dan kemudian dikirimkan sebagai paket IP ke tujuan. Dalam melakukan pengiriman paket, protokol IP bersifat *unreliable*, *connectionless* dan *datagram delivery service*. Saat ini terdapat dua versi dari protokol IPv4 (32 bit) dan IPv6 (128 bit). *Unreliable* berarti protokol IP tidak menjamin datagram yang dikirim pasti sampai di tujuan. Protokol IP hanya berusaha sebaik mungkin untuk membawa datagram sampai ke tujuan. *Connectionless* berarti dalam mengirim paket ke tujuan tidak ada perjanjian terlebih dahulu (*handshake*). *Datagram delivery service* berarti paket data yang dikirim independen terhadap paket data yang lain. Akibatnya jalur yang ditempuh oleh masing-masing paket berbeda satu dengan lainnya.

- **Internet Protokol versi 4 (IPv4)**

Model pengalamatan dalam IPv4 menggunakan 32 bit bilangan biner. Namun untuk mempermudah penulisannya maka setiap delapan bit biner diwakili oleh satu segmen bilangan oktet, sehingga setiap alamat akan memiliki empat buah segmen dari 0.0.0.0 sampai dengan 255.255.255.255 misalnya 202.152.254.254 sehingga total alamat sebesar 2³².

Alamat IPv4 dibagi menjadi dua bagian yaitu alamat jaringan (network address) dan alamat komputer (*host address*). Network address digunakan untuk menunjukkan di jaringan mana komputer berada, sedangkan “host address” menunjukkan komputer tersebut dalam jaringannya tersebut.



Gambar 2. Sistem Pengalamatan IPv4

Untuk meningkatkan efisiensi dan mempermudah administrasi jaringan, maka dalam suatu jaringan yang besar perlu dibagi-bagi ke dalam jaringan yang lebih kecil. Konsep ini sering disebut dengan *subnetwork /subnetting*

- **Internet Protokol versi 6 (IPv6)**

Pada dasarnya IPv6 dikembangkan untuk mengantisipasi kelangkaan *IP address* yang disediakan oleh IPv4. Karena IPv6 ini tidak lagi menggunakan 32 bit biner tetapi 128 bit biner, sehingga alamat yang mampu disediakan yaitu 2128 atau sebesar 3×10^{38} alamat. Selain itu juga dilakukan perubahan dalam penulisannya yaitu 128 bit alamat dipisahkan menjadi masing-masing 16 bit yang tiap bagian dipisahkan dengan ":" dan dituliskan dengan bilangan hexadesimal. Untuk mengetahui letak subnet dari alamat tersebut maka penulisan alamat IPv6 harus mempunyai format :

5AB4:3C12:5412:66DD:CA74:2176:22BB:6C77 / 64

IPv6-Address
Prefix Length

Dimana 64 merupakan jumlah bit yang menunjukkan alamat subnetnya yaitu

5Ab4:3C12:5412:66DD::/64

- **Mekanisme Transisi IPv4 dan IPv6**

Berdasarkan desain yang ada, IPv4 dan IPv6 tidak dapat kompatibel satu sama lain.

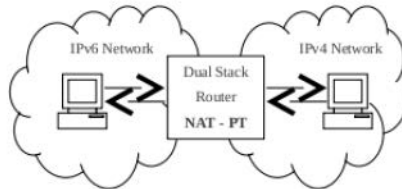
Sehingga diperlukan sebuah mekanisme komunikasi antara Ipv4 dan Ipv6 atau Ipv6 bisa melewati Ipv4. Beberapa jenis transisi yang ada diantaranya *Dual IP Stack*, *Tunneling*, dan *Protocol Translator*

Dual IP Stack, adalah mekanisme transisi yang sederhana, lalu lintas *host* dan *router* dapat dilakukan dengan baik. Gambar 3 menjelaskan bagaimana mekanisme *Dual IP Stack* berlangsung.

Application	
TCP/UDP	
IPv4	IPv6
Network Interface	

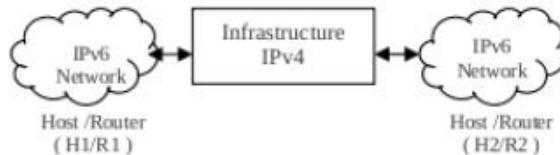
Gambar 3. Dual IP Stack

Address Protocol Translator, gambar 4 menunjukkan proses translasi. Ini memungkinkan *host* dalam Ipv4 berkomunikasi dengan *host* dalam jaringan Ipv6 dan juga *host* Ipv6 berkomunikasi dengan Ipv4 *host*. Dalam hal ini mekanisme *copying*, *translating*, dan penghapusan *header information* terjadi dalam beberapa versi.



Gambar 4. Protocol Translation

Tunneling IPv6 over IPv4 (static), disini menyediakan koneksi IPv6 melalui IPv4 tanpa memodifikasi jaringan IPv4. Paket IPv6 dienkapsulasi ke dalam *header* IPv4 sehingga hal itu akan diperlakukan sebagai paket IPv4. Detail dari mekanisme tersebut dapat dilihat pada gambar 5.



Gambar 5. Tunneling IPv6 over IPv4

IPv6 Tunnel Broker merupakan tempat koneksi user IPv4 untuk melakukan proses registrasi dan aktivasi *tunnel*. *Tunnel broker* bertugas untuk mengatur pembentukan modifikasi dan pembubaran *tunnel* sesuai dengan permintaan *user*. Dalam prakteknya *tunnel broker* dapat membagi beban jaringan kepada beberapa *tunnel server*, dengan cara mengirimkan konfigurasi kepada *tunnel server* yang bersangkutan pada saat *tunnel* tersebut dibentuk, dimodifikasi ataupun dihapus. Selain itu

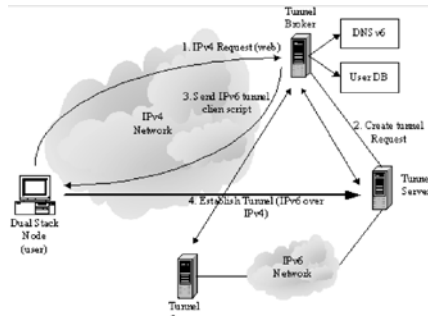
tunnel broker juga berkewajiban untuk mendaftarkan alamat IPv6 *user* dan memasukkannya dalam *DNS server*. *Tunnel broker* harus mendukung IPv4 tetapi tidak harus mendukung IPv6, karena *Tunnel Broker* berhubungan langsung dengan IPv4 dan hubungan *tunnel broker* dan *tunnel server* dapat berupa IPv6 maupun IPv4.

Tunnel Server, merupakan *router dual stack* (IPv4 dan IPv6) yang terhubung dengan jaringan IPv6. *Tunnel server* bertugas menerima seluruh konfigurasi yang dikirim oleh *tunnel broker* pada saat pembangunan, modifikasi dan pembubaran *tunnel disisi server*.

DNS (Domain Name Service) Server, bertugas untuk menerjemahkan (*resolve*) dari nama domain ke alamat IP atau sebaliknya dari pemakai yang telah membentuk *tunnel*. *Server* ini harus mendukung IPv6, karena domain yang kita gunakan merupakan jaringan IPv6.

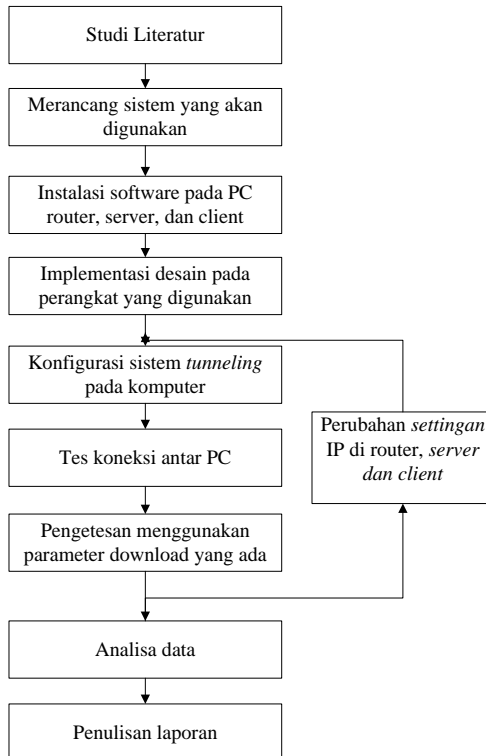
Berikut adalah mekanisme dari ***IPv6 Tunnel Broker*** :

1. *User* menghubungi *tunnel broker* dan dilanjutkan dengan prosedur registrasi (misalnya dengan mengisi form pada web), kemudian *user* akan diberi hak untuk mengakses layanan *tunnel*.
2. *User* menghubungi kembali *tunnel broker*, dan setelah ada proses autentifikasi *user* tersebut memberikan informasi tentang konfigurasi dari host-nya (alamat IP, *Operating System* dan perangkat lunak pendukung IPv6).
3. *Tunnel broker* kemudian mengkonfigurasi *tunnel* di sisi jaringan (*tunnel server*) dan *DNS Server*.
4. Kemudian *user* akan diberikan skrip aktivasi *tunnel* pada sisi *user*. Jika proses ini berhasil maka *user* telah terhubung ke jaringan IPv6 melalui *tunnel server* yang telah ditentukan *tunnel broker*.
5. *User* dapat meminta modifikasi dan pembubaran *tunnel* dengan mengakses *tunnel broker* lagi.



Gambar 6. Mekanisme IPv6 *Tunnel Broker*

G. METODOLOGI



Gambar 7. Diagram metode penelitian

Dari diagram diatas menggambarkan tahapan-tahapan yang akan dilakukan untuk mengerjakan Tugas Akhir ini. Pengumpulan referensi adalah hal pertama yang harus dilakukan untuk memperoleh informasi yang lebih jelas tentang permasalahan yang akan diangkat untuk Tugas Akhir ini. Selanjutnya melakukan suatu perancangan sistem yang akan digunakan.

Tahapan selanjutnya adalah melakukan peng-*install-an Operating Sistem Ubuntu 13.04 Raring Ringtail, Wireshark, dan Apachebench* pada komputer yang akan digunakan sebagai media bantu dalam mengerjakan Tugas Akhir. Kemudian sistem yang sudah dirancang diterapkan pada komputer yang meliputi konfigurasi sistem *tunneling* pada PC *router* dan pengesetan IP *address* pada masing-masing PC tersebut serta melakukan konfigurasi *webserver* dan *ftpserver*.

Setelah semua konfigurasi selesai dilakukan, selanjutnya dilakukan pengetesan koneksi terhadap tiga PC tersebut apakah jaringan yang dibangun sudah saling terkoneksi antar semua PC. Langkah selanjutnya melakukan pengujian pengaruh *tunneling* terhadap *server* dengan menggunakan parameter *download* oleh *client* dengan 5 ukuran paket yang berbeda dan dikirimkan masing-masing dalam 10 waktu tiap paketnya. Dan langkah akhir yang dilakukan dalam tahapan pengerjaan Tugas Akhir ini adalah menuliskan laporan berdasarkan analisa terhadap sistem yang sudah dirancang dan di implementasikan.

H. JADWAL KEGIATAN

Jadwal pelaksanaan kegiatan yang harus dilakukan dapat dilihat pada tabel berikut.

Tabel 1. Jadwal Pelaksanaan Kegiatan

Kegiatan	Bulan I				Bulan II				Bulan III				Bulan IV			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Studi Literatur																
Perancangan Sistem																
Implementasi Sistem																
Pengujian unjuk kerja sistem																
Analisis Hasil Pengujian																
Penulisan buku Tugas Akhir dan <i>Proceeding</i>																

I. DAFTAR PUSTAKA

1. Amoss J. Jonas dan Minoli Daniel, "Handbook of IPv4 to IPv6 Transition Metodologies for Institutional and Corporate Networks", Auerbach Publications, 2008.
2. Aris C, R. Rumani, "IPv6 Tunnel Broker Implementation and Analysis for IPv6 and IPv4 Interconnection", The 6th International Conference on Telecommunication Systems, Services, and Applications, 2011.
3. Ettikan Kandasamy, "Application Performance Analysis in Transition Mechanism from IPv4 to IPv6", Research & Business Development Department, Malaysia.
4. Hinden M. Robert, "IP Next Generation Overview", 1995.
5. Mohammad Syafrudin, "Analisa Unjuk Kerja Routing Protocol Ripng Dan Ospfv3 Pada Jaringan Ipv6, Universitas Indonesia", 2010.
6. Reko Ardonto, "Analisa dan Implementasi Ipv 6 Tunnel Broker untuk Interkoneksi antara Ipv6 dan Ipv4", Universitas Diponegoro.
7. RFC 3053, "IPv6 Tunnel Broker", Internet Society, 2001.
8. Semen, Laurens A.; Wardana, Hartanto Kusuma; Handoko, Implementasi Interkoneksi IPv6 dan IPv4 dengan Menggunakan Mikrotik Router OS Versi 3.15, Jurnal Ilmiah Elektroteknika, Vol. 9, No. 1, 2010.
9. Sukaridhoto S, "Perancangan dan Implementasi jaringan IPv6 di ITS-NET dengan Sistem Operasi Linux", ITS, 2002.

Lampiran B

B.1 Script Home dan Fungsi Tunnel Broker

```
<?php if ( ! defined('BASEPATH')) exit('No direct
script access allowed');
session_start(); //we need to call PHP's session object
to access it through CI
class Home extends CI_Controller {

function __construct()
{
    parent::__construct();
}

function index()
{
    if($this->session->userdata('logged_in'))
    {
        $session_data=$this->session->userdata('logged_in');
        $query = $this->db->get_where('db_user', array('id'
=> $session_data['id']));

foreach($query->result() as $value){
    if($value->ipv4){$ipv4 = $value->ipv4;}else{
        $ipv4 = 'Tidak Ada Data';}
    if($value->ipv6){
        $queryip=$this->db-
        >get_where('db_ipv6', array('id' =>
        $value->ipv6));
        foreach($queryip->result() as $ip){
            $ip6 = $ip->ipv6;
        }
    }else{ $ip6 = 'Tidak Ada Data';}
    if($value->namalengkap){$namalengkap =
    $value->namalengkap;}else{$namalengkap =
    'Tidak ada data';}
    if($value->alamat){$alamat = $value-
    >alamat;}else{$alamat = 'Tidak ada data';}
    if($value->negara){$negara = $value-
    >negara;}else{$negara = 'Tidak ada data';}
    if($value->email){$email = $value-
    >email;}else{$email = 'Tidak ada data';}
    $data = array(
        'title' => 'Home',
        'namalengkap' => $namalengkap,
        'alamat' => $alamat,
        'negara' => $negara,
        'email' => $email,
```

```

        'ipv4' => $ipv4,
        'ipv6' => $ipv6,
    );
}

$this->load->view('head', $data);
$this->load->view('home', $data);
$this->load->view('attribution');
}
else
{
    //If no session, redirect to login page
    redirect('login', 'refresh');
}
}

function logout()
{
    $this->session->unset_userdata('logged_in');
    session_destroy();
    redirect('home', 'refresh');
}

function system()
{
    if($this->session->userdata('logged_in'))
    {
        $session_data          =          $this->session->
        >userdata('logged_in');
        $query = $this->db->get('db_ipv6');

        foreach($query->result() as $value){
            if($value->active==1){$active          =          'Digunakan';}else{          $active          =          'Tidak
            Digunakan';}
            $data = array(
                'title' => 'System',
                'ipv6' => $value->ipv6,
                'ipv6' => $active,
            );
        }

        $this->load->view('head', $data);
        $this->load->view('system', $data);
        $this->load->view('attribution');
    }
    else
    {
        //If no session, redirect to login page
    }
}

```

```

        redirect('login', 'refresh');
    }
}

function config()
{
    if($this->session->userdata('logged_in'))
    {
        $session_data = $this->session->
        >userdata('logged_in');
        $query = $this->db->get_where('db_user', array('id'
        => $session_data['id']));

        foreach($query->result() as $value){
            if($value->ipv4){$ipv4 = $value->ipv4;}else{
                $ipv4 = 'Tidak Ada Data';}
            if($value->ipv6){
                $queryip = $this->db->
                >get_where('db_ipv6', array('id' =>
                $value->ipv6));
                foreach($queryip->result() as $ip){
                    $ipv6 = $ip->ipv6;
                }
                $konfigurasi =
                'configure terminal' . '&#10;' .
                'interface tunnel0' . '&#10;' .
                ' no ip address' . '&#10;' .
                ' ipv6 enable' . '&#10;' .
                ' ipv6 address ' . $ipv6 . '&#10;' .
                ' tunnel source ' . $ipv4 . '&#10;' .
                ' tunnel destination 192.168.0.2' .
                '&#10;' .
                ' tunnel mode ipv6ip' . '&#10;' .
                'ipv6route ::/0 tunnel0' . '&#10;' .
                'end' ;

            }else{ $ipv6 = 'Tidak Ada Data';
                $konfigurasi= '';}
            $data = array(
                'title' => 'Config',
                'ipv4' => $ipv4,
                'ipv6' => $ipv6,
                'konfigurasi' => $konfigurasi
            );
        }

        $this->load->view('head', $data);
        $this->load->view('config', $data);
        $this->load->view('attribution');
    }
}

```

```

    }
    else
    {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

function config_ip()
{
    if($this->session->userdata('logged_in'))
    {
        $session_data      =          $this->session->
        >userdata('logged_in');

        $query_user        =          $this->db->get_where('db_user',
        array('id' => $session_data['id']));

        if($query_user){
            foreach($query_user->result() as $value){
                if($value->ipv4){$ipv4 = $value->ipv4;}else{
                    $ipv4 = 'Tidak Ada Data';}

                    $isactive = array(
                        'active' => 0
                    );

                    $this->db->where('id', $value->ipv6);
                    $this->db->update('db_ipv6',
                    $isactive);
                }
            }

            $query          =          $this->db->get_where('db_ipv6',
            array('active' => 0), 1);

            if($query){
                foreach($query->result() as $value){
                    $ipv6 = $value->ipv6;
                    $idipv6 = $value->id;
                }

                $db = array(
                    'ipv4' => $this->input->post('ipv4'),
                    'ipv6' => $idipv6
                );

                $this->db->where('id', $session_data['id']);
                $this->db->update('db_user', $db);
            }
        }
    }
}

```

```

    $active = array(
        'active' => 1
    );

    $this->db->where('id', $idipv6);
    $this->db->update('db_ipv6', $active);

    $konfigurasi =
        'configure terminal' . '&#10;' .
        'interface tunnel0' . '&#10;' .
        '    no ip address' . '&#10;' .
        '    ipv6 enable' . '&#10;' .
        '    ipv6 address ' . $ipv6 .
        '&#10;' .
        '    tunnel source ' . $ipv4 .
        '&#10;' .
        '    tunnel destination' .
        '192.168.0.2' . '&#10;' .
        '    tunnel mode ipv6ip' .
        '&#10;' .
        'ipv6route ::/0 tunnel0' .
        '&#10;' .
        'end' ;

    $data = array(
        'ipv4' => $this->input->post('ipv4'),
        'ipv6' => $ipv6,
        'konfigurasi' => $konfigurasi
    );
} else {
    $konfigurasi = '';
    $data = array(
        'ipv4' => $this->input->post('ipv4'),
        'ipv6' => 'IPv6 Habis',
        'konfigurasi' => $konfigurasi
    );
}

$title = array('title' => 'Config');

$this->load->view('head', $title);
$this->load->view('config', $data);
$this->load->view('attribution');
}
else
{
    //If no session, redirect to login page
    redirect('login', 'refresh');
}

```

```

}

function config_del()
{
    if($this->session->userdata('logged_in'))
    {
        $session_data = $this->session->userdata('logged_in');
        $query_user = $this->db->get_where('db_user',
        array('id' => $session_data['id']));

        if($query_user){
            foreach($query_user->result() as $value){
                if($value->ipv4){$ipv4 = $value->ipv4;}else{ $ipv4 = 'Tidak Ada Data';}
                $isactive = array(
                    'active' => 0
                );

                $this->db->where('id', $value->ipv6);
                $this->db->update('db_ipv6',
                $isactive);

                $db = array(
                    'ipv6' => ''
                );

                $this->db->where('id',
                $session_data['id']);
                $this->db->update('db_user', $db);

                $konfigurasi = '';
                $data = array(
                    'ipv4' => $ipv4,
                    'ipv6' => 'Tidak Ada Data',
                    'konfigurasi' => $konfigurasi
                );
            }
        }

        $title = array('title' => 'Config');

        $this->load->view('head', $title);
        $this->load->view('config', $data);
        $this->load->view('attribution');
    }
    else

```

```

    {
        //If no session, redirect to login page
        redirect('login', 'refresh');
    }
}

?>

```

B.2 Script Login

```

<?php if ( ! defined('BASEPATH')) exit('No direct
script access allowed');

class Login extends CI_Controller {

    function __construct()
    {
        parent::__construct();
    }

    function index()
    {
        $this->load->helper(array('form'));
        $data = array('title' => 'Login');
        $this->load->view('head', $data);
        $this->load->view('login');
        $this->load->view('attribution');
    }
}

?>

```

B.3 Script Registrasi Tunnel Broker

```

<?php if ( ! defined('BASEPATH')) exit('No direct
script access allowed');

class Daftar extends CI_Controller {

    function __construct()
    {
        parent::__construct();
        $this->load->model('user', '', TRUE);
    }

    function index()
    {
        $data = array('title' => 'Daftar');
    }
}

```



```

$this->load->view('head', $data);
$this->load->view('daftar');
$this->load->view('attribution');
}

function baru(){
    $this->load->helper('security');

    $data = array(
        'username' => $this->input->post('username'),
        'password' => do_hash($this->input-
    >post('password'), 'md5'),
        'namalengkap' => $this->input-
    >post('namalengkap'),
        'alamat' => $this->input->post('alamat'),
        'negara' => $this->input->post('negara'),
        'email' => $this->input->post('email')
    );
    $str = $this->db->insert('db_user', $data);

    $this->load->helper(array('form'));
    $data = array('title' => 'Login');
    $this->load->view('head', $data);

    $notif = array('message' => 'Berhasil Daftar Baru!');
    $this->load->view('notif', $notif);
    $this->load->view('login');
    $this->load->view('attribution');
}

}
?>

```

B.4 Script Verifikasi Login

```

<?php if ( ! defined('BASEPATH')) exit('No direct script
access allowed');

class VerifyLogin extends CI_Controller {

    function __construct()
    {
        parent::__construct();
        $this->load->model('user', '', TRUE);
    }

    function index()
    {
        //This method will have the credentials validation
        $this->load->library('form_validation');
    }
}

```

```

    $this->form_validation-
>set_rules('username','Username',
'trim|required|xss_clean');
    $this->form_validation-
>set_rules('password','Password',
'trim|required|xss_clean|callback_check_database');

    if($this->form_validation->run() == FALSE)
    {
        //Field validation failed. User redirected to
login page
        $this->load->helper(array('form'));
        $data = array('title' => 'Login');
        $this->load->view('head', $data);
        $this->load->view('login');
        $this->load->view('attribution');
    }
    else
    {
        //Go to private area
        redirect('home', 'refresh');
    }
}

function check_database($password)
{
    //Field validation succeeded. Validate against
database
    $username = $this->input->post('username');

    //query the database
    $result = $this->user->login($username, $password);

    if($result)
    {
        $sess_array = array();
        foreach($result as $row)
        {
            $sess_array = array(
                'id' => $row->id,
                'username' => $row->username
            );
            $this->session->set_userdata('logged_in',
$sess_array);
        }
        return TRUE;
    }
}

```

```

else
{
    $this->form_validation-
>set_message('check_database','Invalid      username      or
password');
    return false;
}
}
}
?>

```

B.5 Command Tunnel Broker

```

%routerlipv4
Router>enable
Router#configure terminal
Router(config)#interface fa0/0
Router(config-if)#ip address 10.10.10.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fa0/1
Router(config-if)#ip address 192.168.0.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#exit
Router#

%routerlipv6
Router>enable
Router#configure terminal
Router(config)#interface Tunnel0
Router(config-if)#no ip address
Router(config-if)#ipv6 enable
Router(config-if)#ipv6 address 2001:B::1/64
Router(config-if)#tunnel source 192.168.0.1
Router(config-if)#tunnel destination 192.168.0.2
Router(config-if)#tunnel mode ipv6ip
Router(config-if)#exit
Router(config)#interface fa0/1
Router(config-if)#ipv6 address 2001:A::1/64
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#ipv6 route ::0 Tunnel0
Router#

%routertunnelbroker
#!b/bin/bash
ifconfig eth0 up
ifconfig eth0 inet add 192.168.0.2 netmask 255.255.255.0
ifconfig eth0 inet6 add 2001:b::2/64

```

```
endpoint 192.168.0.1
local 192.168.0.2
gateway 2001:b::1/64
ifconfig eth1 up
ifconfig eth1 inet6 add 2001:c::1/64
```

B.6 Command IPv4

```
%router1
Router>enable
Router#configure terminal
Router(config)#interface fa0/0
Router(config-if)#ip address 10.10.10.2 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fa0/1
Router(config-if)#ip address 192.168.0.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#ip route 0.0.0.0 0.0.0.0 192.168.0.2
Router#
```

```
%router2
#!/bin/bash
ifconfig eth0 up
ifconfig eth0 inet 192.168.0.2 netmask 255.255.255.0
ifconfig eth1 up
ifconfig eth1 inet 202.154.0.1 netmask 255.255.255.0
echo "1">/proc/sys/net/ipv4/ip_forward
route add -net 192.168.0.0 netmask 255.255.255.0 gw
192.168.0.1
```

B.7 Command Router 1 IPv6

```
%router1ipv4
Router>enable
Router#configure terminal
Router(config)#interface fa0/0
Router(config-if)#no ip address
Router(config-if)#ipv6 address 2001:0:0:a::2/64
Router(config-if)#ipv6 enable
Router(config-if)#ipv6 rip process1 enable
Router(config-if)#exit
Router(config)#interface fa0/1
Router(config-if)#ipv6 address 2001:0:0:b::2/64
Router(config-if)#ipv6 enable
Router(config-if)#ipv6 rip process1 enable
Router(config-if)#exit
Router(config)#exit
%router2
```

```
#!/bin/bash
ifconfig eth0 up
ifconfig eht0 inet6 2001:0:0:0:b::1/64
ifconfig eth1 up
ifconfig eht1 inet 2001:0:0:0:c::1/64
echo"1">/proc/sys/net/ipv4/ip_forward
route -A inet6 2001:0:0:c:: gw 2001:0:0:0:b::2/64
```